CONNEXIONS™

The Interoperability Report

August 1990 | *Special Issue: Network Management and Network Security* | Volume 4, No. 8

ConneXions —
The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop, Inc., 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779.

ISSN 0894–5926

From the Editor

The trouble with a term like "Network Management" is that it means different things to different people. Network Management, as discussed in *this* special issue, could perhaps be defined as "the operational aspects of running a network or internetwork." As you may recall, our March 1989 issue was *also* devoted to Network Management, but this time we are not concerned so much with the protocol (wars) as we are with practical hints for network managers. Most of the articles in this issue are written by people directly involved in the operation of networks, and by users of such networks.

Our main article, entitled "A Practical Introduction to Network Management" contains a list of the many pitfalls facing a network manager, and suggestions for how to deal with such issues as configuration, monitoring, fault isolation, capacity planning, and last, but not least, *security*. Security is of course closely related to network management and we will return to this topic with several articles later in this issue.

Marshall Rose has been actively involved in the Internet network management protocol development. His perspective will soon be reflected in his book *The Simple Book: An Introduction to Management of TCP/IP-based internets*. In this issue he gives a brief history of how the Internet network management effort has proceeded over the last few years.

Following the historic overview, we turn to a practical look at how The *Simple Network Management Protocol* (SNMP) can be used to monitor and control an operational internet. SNMP is currently the management protocol of choice for the TCP/IP protocol suite. The article is by Mark Fedor, one of the original authors of SNMP.

Network security is covered in several articles: First, Dr. James M. Galvin of Trusted Information Systems describes the OSI Security Architecture in our series *Components of OSI*. This is followed by two *User Viewpoints*, one on the IP security option, and one on the use of IP packet filtering. The articles are by David Wiltzius of Lawrence Livermore National Laboratories, and Richard Kent of Network Systems Corporation respectively.

Eileen Forrester gives an overview of the *Computer Emergency Response Team* (CERT) which was set up after the infamous *Internet Worm* incident of November 1988. Following her article are some hints for network managers, taken from CERT Advisory bulletins.

Finally, a couple of pages with information about where to learn more, including a UNIX Security white paper, FIPS security documents, a mailing list for Privacy Enhanced Mail, references to previous articles in *ConneXions*, and some upcoming events.

A Practical Introduction to Network Management

by
Robert H. Stine, Applied Cybernetics, Inc.
J. Paul Holbrook, CERT
Michael A. Patton, MIT
James B. VanBokkelen, FTP Software

Introduction

This tutorial is a practical guide for network management at sites using the TCP/IP protocol suite. It addresses system monitoring, fault detection and isolation, performance management, configuration management, and Internet security. It closes with the ultimate advice for hard-pressed network managers.

We assume an acquaintance with the TCP/IP protocol suite and the Internet architecture. There are many available references on these topics, several of which are listed below.

Reading this tutorial is no substitute for knowing your system, and knowing how it is used. Do not wait until something breaks to learn what your system ought to do or how it usually works: a crisis is not the time determine how "normal" packet traces should look. Furthermore, it takes little imagination to realize that you do not want to be digging through manuals while your boss is screaming for network service to be restored.

Since many of the details of network management are system-specific, this tutorial is a bit broad-brush and high-level in its presentation. In addition to system diversity, however, there is a more fundamental problem in prescribing network management practices: network management is not a well-understood endeavor. At present, the cutting edge of network management is the use of distributed systems to collect and exchange status information, and then to display the data as histograms or trend lines. It is not clear that we know what data should be collected, how to analyze it when we get it, or how to structure our collection systems. For now, automated, real-time control of internets is an aspiration, rather than a reality. The communications systems that we routinely field are apparently more complex than we can comprehend, which no doubt accounts in part for their frequently surprising behavior.

Goals and functions

An organization's view of network management goals is shaped by two factors:

- The organization depends on the system working; and
- LANs, routers, lines, and other communications resources have costs.

From the organizational vantage point, the ultimate goal of network management is to provide a consistent, predictable, acceptable level of service from the available data communications resources. To achieve this, a network manager must first be able to perform fault detection, isolation, and correction.

People actually managing networks have a different focus. Network managers are usually evaluated by the availability and performance of their communications systems, even though many factors of network performance are beyond their control. To them, the most important requirement of a network management tool is that it allows the detection and diagnosis of faults before users can call to complain: users (and bosses) can often be placated just by knowing that a network problem has been diagnosed.

Other vital network management functions include the ability to effect configuration changes with a minimum of disruption, and to measure the utilization of system components. System measurement can be used to support rational planning for growth, and to justify current or future expenditures for the data communications plant and staff.

System monitoring

System monitoring is a fundamental aspect of network management. One can divide system monitoring into two rough categories: *error detection* and *baseline monitoring*.

System errors, such as misformatted frames or dropped packets, are not in themselves cause for concern. Spikes in error rates, however, should be investigated. It is sound practice to log error rates over time, so that increases can be recognized. Furthermore, logging error rates as a function of traffic rates can be used to detect congestion. Investigate unusual error rates and other anomalies as they are detected, and keep a notebook to record your discoveries.

Day-to-day traffic should be monitored, so that the operational baselines of a system and its components can be determined. As well as being essential for performance management, baseline determination and traffic monitoring are the keys to early fault detection.

A preliminary step to developing baseline measurements is construction of a system map: a graphical representation of the system components and their interfaces. Then, measurements of utilization (i.e., use divided by capacity) are needed. Problems are most likely to arise, and system tuning efforts are most likely to be beneficial, at highly utilized components.

It is worthwhile to develop a source/destination traffic matrix, including a breakdown of traffic between the local system and other internet sites. Both volume and type of traffic should be logged, along with its evolution over time. Of particular interest for systems with diskless workstations is memory swapping and other disk server access. For all systems, broadcast traffic and routing traffic should be monitored. Sudden increases in the variance of delay or the volume of routing traffic may indicate thrashing or other soft failures.

In monitoring a system, long-term averages are of little use. Hourly averages are a better indicator of system use. Variance in utilization and delay should also be tracked. Sudden spikes in variance are telltale signs that a problem is looming or exists. So, too, are trends of increased packet or line errors, broadcasts, routing traffic, or delay.

Fault detection and isolation

When a system fails, caution is in order. A network manager should make an attempt to diagnose the cause of a system crash before rebooting. In many cases, however, a quick diagnosis will not be possible. For some high priority applications, restoring at least some level of service will have priority over fault repair or even complete fault diagnosis. This necessitates prior planning. A network manager must know the vital applications at his site. If applications require it, he must also have a fall-back plan for bringing them online. Meanwhile, repeated crashes or hardware failures are unambiguous signs of a problem that must be corrected.

A network manager should prepare for fault diagnosis by becoming familiar with how diagnostic tools respond to network failure. During peacful times, he should occasionally unplug the network connection from an unused workstation and then "debug" the problem.

continued on next page

Practical Intro to Network Management *(continued)*

When diagnosing a fault or anomaly, it is vital to proceed in an orderly manner, especially since network faults will usually generate spurious as well as accurate error messages. Remember to keep in mind that the network itself is failing. Do not place too much trust in anything obtained remotely. Furthermore, it is unlikely to be significant that remote information such as DNS names or NFS files cannot be obtained.

Even spurious messages can be revealing, because they provide clues to the problem. From the data at hand, develop working hypotheses about probable causes of the problems you detect. Direct your further data gathering efforts so that the information you get will either refute or support your hypotheses.

An orderly approach to debugging is facilitated if it is guided by a model of network behavior. The following portions of this section present such a model, along with a procedure for checking network connectivity. The section concludes with some hints for diagnosing a particularly tricky class of connectivity problem.

A network model as a diagnostic framework

The point of having a model of how things work is to have a basis for developing educated guesses about how things go wrong. The problem of *cascading faults*—faults generating other faults—makes use of a conceptual model a necessity.

In general, only problems in a component's hardware or operating system will generate simultaneous faults in multiple protocol layers. Otherwise, faults will propagate vertically (up the protocol stack) or horizontally (between peer-level communications components). Applying a conceptual model that includes the architectural relations of network components can help to order an otherwise senseless barrage of error messages and symptoms. The model does not have to be formal or complex to bring structure to debugging efforts. A useful start is something as simple as the following:

• *Applications programs use TCP/UDP transport services*: Before using service, applications that accept host names as parameters must translate the names into IP addresses. Translation may be based on a static table lookup (`/etc/hosts` file in UNIX hosts), the DNS, or YP. *Nslookup* and *DiG* are tools for monitoring the activities of the DNS.

• *Transport protocol implementations use IP services:* The local IP module makes the initial decision on forwarding. An IP datagram is forwarded directly to the destination host if the destination is on the same network as the source. Otherwise, the datagram is forwarded to a gateway attached to the network. On BSD hosts, the contents of a host's routing table are visible by use of the *netstat* command. (Initial forwarding may actually be complex and vulnerable to multiple points of failure. For example, when sending an IP datagram, 4.3BSD hosts first look for a route to the particular host. If none has been specified for the destination, then a search is made for a route to the network of the destination. If this search also fails, then as a last resort, a search is made for a route to a "default" gateway. Routes to hosts, networks, and the "default" gateway may be static, loaded at boot time and perhaps updated by operator commands. Alternatively, they may be dynamic, loaded from redirects and routing protocol updates.)

• *IP implementations translate the IP address of a datagram's next hop:* (either the destination host or a gateway) to a local network address. For Ethernets, the *Address Resolution Protocol* (ARP) is commonly used for this translation. On BSD systems, an interface's IP address and other configuration options can be viewed by use of the *ifconfig* command, while the contents of a host's ARP cache may be viewed by use of the *arp* command.

• *IP implementations in hosts and gateways route datagrams based on subnet and net identifiers:* Subnetting is a means of allocating and preserving IP address space, and of insulating users from the topological details of a multi-network campus. Sites that use subnetting reserve portions of the IP address's host identifier to indicate particular networks at their campus. Subnetting is highly system-dependent. The details are a critical, though local, issue. As for routing between separate networks, a variety of gateway-to-gateway protocols are used. *traceroute* is a useful tool for investigating routing problems. The tool, *query,* can be used to examine RIP routing tables.

A network manager should expand the above description so that it accurately describes his particular system, and learn the tools and techniques for monitoring the operations at each of the above stages.

Simple procedure for connectivity check

In this section, we describe a procedure for isolating a TCP/IP connectivity problem. In this procedure, a series of tests methodically examine connectivity from a host, starting with nearby resources and working outward. The steps in our connectivity-testing procedure are:

1. As an initial sanity check, *ping* your own IP address and the loopback address.

2. Next, try to *ping* other IP hosts on the local subnet. Use numeric addresses when starting off, since this eliminates the name resolvers and host tables as potential sources of problems. The lack of an answer may indicate either that the destination host did not respond to ARP (if it is used on your LAN), or that a datagram was forwarded (and hence, the destination IP address was resolved to a local media address) but that no ICMP Echo Reply was received. This could indicate a length-related problem, or mis-configured IP Security.

3. If an IP router (gateway) is in the system, *ping* both its near and far-side addresses.

4. Make sure that your local host recognizes the gateway as a relay. (For BSD hosts, use *netstat.*)

5. Still using numeric IP addresses, try to *ping* hosts beyond the gateway. If you get no response, run *hopcheck* or *traceroute*, if available. Note whether your packets even go to the gateway on their way to the destination. If not, examine the methods used to instruct your host to use this gateway to reach the specified destination net (e.g., is the default route in place? Alternatively, are you successfully wire-tapping the IGP messages broadcast on the net you are attached to?)

If *traceroute* is not available, *ping, netstat, arp,* and a knowledge of the IP addresses of all the gateway's interfaces can be used to isolate the cause of the problem. Use *netstat* to determine your next hop to the destination. *Ping* that IP address to ensure the router is up. Next, ping the router interface on the far subnet. If the router returns "network unreachable" or other errors, investigate the router's routing tables and interface status.

continued on next page

Practical Intro to Network Management *(continued)*

If the *pings* succeed, *ping* the close interface of the succeeding next hop gateway, and so on. Remember the routing along the outbound and return paths may be different.

6. Once *ping* is working with numeric addresses, use *ping* to try to reach a few remote hosts by name. If *ping* fails when host names are used, check the operation of the local name-mapping system (i.e., with *nslookup* or *DiG*). If you want to use "shorthand" forms ("`myhost`" instead of "`myhost.mydomain.com`"), be sure that the alias tables are correctly configured.

7. Once basic reachability has been established with *ping*, try some TCP-based applications: FTP and Telnet are supported on almost all IP hosts, but Finger is a simpler protocol.

The Berkeley-specific protocols *(rsh, rcp, rexec* and *lpr)* require extra configuration on the server host before they can work, and so are poor choices for connectivity testing.

If problems arise in steps 2–7 above, rerunning the tests while executing a line monitor (e.g., *etherfind, netwatch,* or *tcpdump*) can help to pinpoint the problem.

The above procedure is sound and useful, especially if little is known about the cause of the connectivity problem. It is not, however, guaranteed to be the shortest path to diagnosis. In some cases, a binary search on the problem might be more effective (i.e., try a test "in the middle," in a spot where the failure modes are well defined). In other cases, available information might so strongly suggest a particular failure that immediately testing for it is in order. This last "approach," which might be called "hunting and pecking," should be used with caution: chasing one will o' the wisp after another can waste much time and effort.

Note that line problems are still among the most common causes of connectivity loss. Problems in transmission across local media are outside the scope of this Tutorial. But, if a host or workstation loses or cannot establish connectivity, check its physical connection.

Limited connectivity

An interesting class of problems can result in a particularly mysterious failure: Telnet or other low-volume TCP connections work, but large file transfers fail. FTP transfers may start, but then hang. There are several possible culprits in this problem. The most likely suspects are IP implementations that cannot fragment or reassemble datagrams, and TCP implementations that do not perform dynamic window sizing (a.k.a. Van Jacobson's "Slow Start" algorithm). Another possibility is mixing incompatible frame formats on an Ethernet.

Even today, some IP implementations in the Internet cannot correctly handle fragmentation or reassembly. They will work fine for small packets, but drop all large packets.

Buffer exhaustion

The problem can also be caused by buffer exhaustion at gateways that connect interfaces of widely differing bandwidth. Datagrams from a TCP connection that traverses a bottleneck will experience queue delays, and will be dropped if buffer resources are depleted.

The congestion can be made worse if the TCP implementation at the traffic source does not use the recommended algorithms for computing retransmission times, since spuriously retransmitted datagrams will only add to the congestion. [To avoid this problem, TCP implementations on the Internet must use "exponential backoff" between successive retransmissions, Karn's algorithm for filtering samples used to estimate round-trip delay between TCP peers, and Jacobson's algorithm for incorporating variance into the "retransmission time-out" computation for TCP segments. See Section 4.2.3.1 of RFC 1122].

Fragmentation, even if correctly implemented, will compound this problem, since processing delays and congestion will be increased at the bottleneck.

Serial Line Internet Protocol (SLIP) links are especially vulnerable to this and other congestion problems. SLIP lines are usually an order of magnitude slower than other gateway interfaces.

MTU size

Also, the SLIP lines are at times configured with MTUs (*Maximum Transmission Unit*), the maximum length of an IP datagram for a particular subnet) as small as 256 bytes, which virtually guarantees fragmentation. To alleviate this problem, TCP implementations behind slow lines should advertise small windows. Also, if possible, SLIP lines should be configured with an MTU no less than 576 bytes. The tradeoff to weigh is whether interactive traffic will be penalized too severely by transmission delays of lengthy datagrams from concurrent file transfers.

Trailers

Misuse of Ethernet trailers can also cause the problem of hanging file transfers. "Trailers" refers to an Ethernet frame format optionally employed by BSD systems to minimize buffer copying by system software. BSD systems with Ethernet interfaces can be configured to send large frames so that their address and control data are at the end of a frame (hence, a "trailer" instead of a "header"). After a memory page is allocated and loaded with a received Ethernet frame, the Ethernet data will begin at the start of the memory page boundary. Hence, the Ethernet control information can be logically stripped from the end merely by adjusting the page's length field. By manipulating virtual memory mapping, this same page (sans Ethernet control information), can then be passed to the local IP module without additional allocation and loading of memory. The disadvantage in using trailers is that it is non-standard. Many existing implementations cannot parse trailers.

The hanging FTP problem will appear if a gateway is not configured to recognize trailers, but a host or gateway immediately "upstream" on an Ethernet uses them. Short datagrams will not be formatted with trailers, and so will be processed correctly. When the bulk data transfer starts, however, full-sized frames will be sent, and will use the trailer format. To the gateway that receives them, they appear simply as misformatted frames, and are quietly dropped. The solution, obviously, is to insure that all hosts and gateways on an Ethernet are consistent in their use of trailers. Note that RFC 1122, "Requirements for Internet Host," places severe restrictions on the use of trailers.

Performance management

Performance management encompasses two rather different activities. One is passive system monitoring to detect problems and determine operational baselines.

continued on next page

Practical Intro to Network Management *(continued)*

The goal is to measure system and component utilization and so locate bottlenecks, since bottlenecks should receive the focus of performance tuning efforts. Also, performance data is usually required by upper level management to justify the costs of communications systems. This is essentially identical to system monitoring, and is addressed at greater length above.

Capacity planning

Another aspect of performance management is active performance testing and capacity planning. Some work in this area can be based on analysis. For example, a rough estimate of gateway capacity can be deduced from a simple model given by Charles Hedrick in his "Introduction to Administration of an Internet-based Local Network," which is:

per-packet processing time =
switching time + (packet size) * (transmission bps).

Another guideline for capacity planning is that in order to avoid excessive queuing delays, a system should be sized at about double its expected load. In other words, system capacity should be so high that utilization is no greater than 50%.

Although there are more sophisticated analytic models of communications systems than those above, their added complexity does not usually gain a corresponding accuracy. Most analytic models of communications nets require assumptions about traffic load distributions and service rates that are not merely problematic, but are patently false. These errors tend to result in underestimating queuing delays. Hence, it is often necessary to actually load and measure the performance of a real communications system if one is to get accurate performance predictions. Obviously, this type of testing is performed on isolated systems or during off hours. The results can be used to evaluate parameter settings or predict performance during normal operations.

Simulation

Simulations can be used to supplement the testing of real systems. To be believable, however, simulations require validation, which, in turn, requires measurements from a real system. Whether testing or simulating a system's performance, actual traffic traces should be incorporated as input to traffic generators. The performance of a communications system will be greatly influenced by its load characteristics (burstiness, volume, etc.), which are themselves highly dependent on the applications that are run.

When tuning a net, in addition to the usual configuration parameters, consider the impact of the location of gateways and print and file servers. A few rules of thumb can guide the location of shared system resources. First, there is the principle of locality: a system will perform better if most traffic is between nearby destinations. The second rule is to avoid creating bottlenecks. For example, multiple diskservers may be called for to support a large number of workstations. Furthermore, to avoid LAN and diskserver congestion, workstations should be configured with enough memory to avoid frequent swapping.

As a final note on performance management, proceed cautiously if your Ethernet interface allows you to customize its collision recovery algorithm. This is almost always a bad idea.

The best that it can accomplish is to give a few favored hosts a disproportionate share of the Ethernet bandwidth, perhaps at the cost of a reduction in total system throughput. Worse, it is possible that differing collision recovery algorithms may exhibit a self-synchronizing behavior, so that excess collisions are generated.

Configuration management

Configuration management is the setting, collecting, and storing of the state and parameters of network resources. It overlaps all other network management functions. Hence, some aspects of configuration management have already been addressed (e.g., tuning for performance). In this section, we will focus on configuration management activities needed to "hook up" a net or campus to a larger internet. We will not, of course, include specific details on installing or maintaining internetted communications systems. We will, however, skim over some of the TCP/IP configuration highlights.

Configuration management includes "name management"—the control and allocation of system names and addresses, and the translation between names and addresses. Name-to-address translation is performed by "name servers." We conclude this section with a few strictures on the simultaneous use of two automated name-servers, the Domain Name System (DNS), and Yellow Pages (YP).

Required host configuration data for TCP/IP internets

In a TCP/IP internet, each host needs several items of information for internet communications. Some will be host-specific, while other information will be common for all hosts on a subnet. In a draft document, ["Dynamic Configuration of Internet Hosts"] Ralph Droms identifies the following configuration data required by internet hosts:

- An IP address, a host specific value that can be statically configured or obtained via BOOTP, the *Reverse Address Resolution Protocol* (RARP) or *Dynamic RARP* (DRARP).

- Subnet properties, such as the subnet mask and the Maximum Transmission Unit (MTU); obviously, these values are not host-specific.

- Addresses of "entry" gateways to the internet; addresses of default gateways are usually statically configured; though the ICMP "redirect" message can be used to refine a host's routing tables, there is currently no dynamic TCP/IP mechanism or protocol for a host to locate a gateway; an IETF working group is busy on this problem.

- For hosts in internets using the Domain Name System (DNS) for name-to-address translation, the location of a local DNS server is needed; this information is not host-specific, and usually statically configured;

- Host name (domain name, for hosts using DNS); obviously host-specific; either hard-coded or obtained in a boot procedure.

- For diskless hosts, various boot services. BOOTP is the standard Internet protocol for downloading boot configuration information. The *Trivial File Transfer Protocol* (TFTP) is typically used for downloading boot images. Sun computers use the `bootparams` RPC mechanism for downloading initial configuration data to a host.

continued on next page

Practical Intro to Network Management *(continued)*

There are ongoing developments, most notably the work of the *Dynamic Host Configuration Working Group* of the IETF, to support dynamic, automatic gathering of the above data. In the meantime, most systems will rely on hand-crafted configuration files. [Ed.: See "Automatic Configuration of Internet Hosts," *ConneXions*, Volume 4, No. 3, March 1990.]

IP address format

An IP address consists of a network identifier, an optional subnet identifier, and a host identifier. None of these fields can be assigned arbitrarily.

Internet routing is based on the network identifier. Separate, partitioned nets advertising the same IP network number will cause routing chaos. To avoid collisions in network identifiers, selection must be coordinated with the internet administration.

If used, the subnet identifier of an IP address must be at least two bits long. The subnet identifier usually occupies contiguous bits, though this is not necessary.

When initially configuring a network, some thought should be given to how many subnets will eventually be needed. There is a natural tendency to underestimate the growth of a communications system.

On a single net, each host must have a unique host identifier. Another constraint on address assignment is that the use of all 0's as the host number in an IP address is a bad idea. It may work in limited cases, but it is contrary to the specifications, and will fail if pushed.

Default gateways

Each host needs router information, at least to include a default gateway. Gateways need to be configured to use the appropriate route protocol (e.g., RIP, EGP). *gated* is a flexible, though complex process that can simultaneously run RIP, Hello, and EGP, and soon will be able to run BGP and OSPF. If you plan to run *gated* consult with Mark Fedor (`fedor@psi.com`).

Broadcast address

Special care should be taken to ensure that all system components share a common format for IP broadcast addresses. Historically, there have been several broadcast formats. Unfortunately, disaster, in the form of broadcast storms, can result if broadcast addresses are mistaken for specific destinations and are forwarded. For this and other reasons, a host with a single IP interface should *not* be configured to relay packets.

Connecting to *The* Internet

The original TCP/IP Internet (spelled with an upper-case "I") is still active, and still growing. An interesting aspect of the Internet is that it spans many independently administered systems.

Connection to the Internet requires: a registered network number, for use in IP addresses; a registered Autonomous System Number (ASN), for use in internet routing; and, a registered domain name. Fielding a primary and backup DNS server is a condition for registering a domain name.

The *Defense Data Network (DDN) Network Information Center* (NIC) is responsible for registering network numbers, Autonomous System Numbers, and domain names. Regional nets will have their own policies and requirements for Internet connections, but all use the NIC for this registration service.

Contact the NIC at:

DDN Network Information Center
SRI International, Room EJ291
333 Ravenswood Avenue
Menlo Park, CA 94025
E-mail: Hostmaster@nic.ddn.mil
Phone: 1-415-859-3695 1-800-235-3155 (toll-free hotline)

YP and DNS: Dueling name servers

The *Domain Name System* (DNS) provides name service: it translates host names into IP addresses (this mapping is also called "resolution"). Two widespread DNS implementations are *bind* and *named*. The Sun *Yellow Pages* (YP) system can be configured to provide an similar service, through providing remote, keyed access to the hosts.byname map. If both DNS and the YP hosts.byname map are installed, they can interact in disruptive ways.

The problem has been noted in systems in which DNS is used as a fallback, to resolve hostnames that YP cannot. If DNS is slow in responding, the timeout in program *ypserv* may expire, which triggers a repeated request.

Disaster

This can result in disaster if DNS was initially slow because of congestion: the slower things get, the more requests are generated, which slows things even more. A symptom of this problem is that failures by the DNS server or network will trigger numerous requests to DNS.

Reportedly, the bug in YP that results in the avalanche of DNS requests has been repaired in SunOS 4.1. The problem, however, is more fundamental than an implementation error. The YP map hosts.byname and the DNS contain the same class of information. One can get an answer to the same query from each system. These answers may well be different: there is not a mechanism to maintain consistency between the systems. More critical, however, is the lack of a mechanism or procedure to establish which system is authoritative. Hence, running the DNS and YP name services in parallel is pointless. If the systems stay consistent, then only one is needed. If they differ, there is no way to choose which is correct.

Don't run both!

The YP hosts.byname service and DNS are comparable, but incompatible. If possible, a site should not run both services. Because of Internet policy, sites with Internet connections *must* use the DNS. If YP is also used, it should be configured with YP-to-DNS disabled.

Fix

Hacking a system so that it uses DNS rather than the YP hosts.byname map is not trivial, and should not be attempted by novices. The approach is to rebuild the shared C link-library, so that system calls to *gethostbyname()* and *gethostbyaddr()* will use DNS rather than YP. To complete the change, programs that do not dynamically link the shared C library (*rcp, arp,* etc.) must also be rebuilt.

Modified shared C libraries for Sun 3s and Sun 4s are available via anonymous FTP from host uunet.uu.net, in the sun-fixes directory. Note that use of DNS routines rather than YP for general name resolution is not a supported SunOS feature at this time.

Internet security

The guidelines and advice in this section pertain to enhancing the protection of data that is merely "sensitive." By themselves, these measures are insufficient for protecting "classified" data.

continued on next page

Practical Intro to Network Management *(continued)*

Implementing the policies required to protect classified data is subject to stringent, formal review procedures, and is regulated by agencies such as the *Defense Investigative Service* (DIS) and the *National Security Agency* (NSA).

A network manager must realize that he is responsible for protecting his system and its users. Furthermore, though the Internet may appear to be a grand example of a cooperative joint enterprise, recent incidents have made it clear that not all Internet denizens are benign.

A network manager should be aware that the network services he runs have a large impact on the security risks to which his system is exposed. The prudent network manager will be very careful as to what services his site provides to the rest of the Internet, and what access restrictions are enforced. For example, the protocol "finger" may provide more information about a user than should be given to the world at large. Worse, most implementations of the the protocol TFTP give access to all world-readable files.

This section highlights several basic security considerations for Internet sites. It then lists several sources of information and advice on improving the security of systems connected to the Internet.

Basic security

Two major Internet security threats are *denial of service* and *unauthorized access*. Denial of service threats often take the form of protocol spoofers or other malicious traffic generators. These problems can be detected through system monitoring logs. If an attack is suspected, immediately contact your regional net office (e.g., SURANET, MILNET). In addition, DDN users should contact SCC, while other Internet users should contact CERT (see below). A cogent description of your system's symptoms will be needed.

At your own site, be prepared to isolate the problems (e.g., by limiting disk space available to the message queue of a mail system under attack). As a last resort, coping with an attack may require taking down an Internet connection. It is better, however, not to be too quick to quarantine your site, since information for coping with the attack may come via the Internet.

Unauthorized access is a potentially more ominous security threat. The main avenues are attacks against passwords and attacks against privileged system processes. Many system services assume mutual trust, including *rsh, rcp, rlogin,* and NFS *mount*. These are often configured with gaping security holes.

***r*–protocols**

The holes in the "r-protocols" and NFS are principally due to use of an IP address as a credential. The worst threat is from systems on the same cable as the trusting host, because normally one cannot pretend to be a local host from far away—the routers will send the returned data stream into the bit bucket. If an attacker has control of off-network routing, however, the scope of the risk is broadened. Furthermore, an NFS file system can be mis-configured to grant "root" privilege for anonymous *mount* requests, with the result that file systems could be accessed or even destroyed by any remote system with IP connectivity.

Initial passwords

An appallingly common means of gaining entry to systems is by use of the initial passwords to root, sysdiag, and other management accounts that systems are shipped with.

Only slightly less vulnerable are common or trivial passwords, since these are readily subverted by dictionary attacks. (Exotic fantasy creatures and women's names are well represented in most password dictionaries). Obvious steps can reduce the risk of password attacks: passwords should be short-lived, at least eight characters long, with a mix of upper and lower case, and preferably random. The distasteful aspect of memorizing a random string can be alleviated if the password is pronounceable. Special case characters can also be used to good effect.

Improving passwords does not remove all risks. Passwords transmitted over an Ethernet are visible to all attached systems. Furthermore, gateways have the potential to intercept passwords used by any FTP or Telnet connection that traverses them. It is a bad idea for the root account to be accessed by FTP or Telnet if the connections must cross untrusted elements.

System programs

Attacks against system processes are another avenue of unauthorized access. The principle is that by subverting a system process, the attacker can then gain its access privileges.

One approach to reducing this risk is to make system programs harder to subvert. For example, the widespread attack in November 1988 by a self-replicating computer program ("worm," analogous to a tapeworm) subverted the *fingerd* process, by loading an intrusive bootstrap program (known variously as a "grappling hook" or "vector" program), and then corrupting the stack space so that a subroutine's return address was overwritten with the address of the bootstrap program.

An early account of the Internet Worm incident of November 1988 is given by Eugene Spafford in the January 1989 issue of *Computer Communications Review*. Several other articles on the worm incident are in the June 1989 issue of the *Communications of the ACM*.

The security hole in *fingerd* consisted of an input routine that did not have a length check. Security fixes to *fingerd* include the use of a revised input routine.

A more general protection is to apply the principle of "least privilege." Where possible, system routines should run under separate user IDs, and should have no more privilege than is necessary for them to function.

To further protect against attacks on system processes, system managers should regularly check their system programs to ensure that they have not been tampered with or modified in any way. Checksums should be used for this purpose. Using the operating system to check a file's last date of modification is insufficient, since the date itself can be compromised. Finally, to avoid the unauthorized replacement of system code, care should be exercised in assigning protection to its directory paths.

Trap doors

Some system programs actually have "trap doors" that facilitate subversion. A trap door is the epitome of an undocumented feature: it is a hidden capability of a system program that allows a knowledgeable person to gain access to a system. The Internet Worm exploited what was essentially a trap door in the BSD *sendmail* program.

continued on next page

Practical Intro to Network Management *(continued)*

Ensuring against trap doors in software as complex as *sendmail* may be infeasible. In an ideal world, the BSD *sendmail* program would be replaced by an entire mail sub-system (i.e., perhaps including mail user agents, mail transfer agents, and text preparation and filing programs). Any site using *sendmail* should at least obtain the less vulnerable, toughened distribution from `ucbarpa.berkeley.edu`, in the file `~ftp/4.3/sendmail.tar.Z`. Sites running SunOS should note that the 4.0.3 release closed the security holes exploited by the Internet Worm. Fixes for a more obscure security hole in SunOS are available from host `uunet.uu.net` in `~ftp/sun-fixes`; these improvements have been incorporated in SunOS 4.1.

MMDF

Sendmail has problems other than size and complexity. Its use of root privileges, its approach to alias expansion, and several other design characteristics present potential avenues of attack. For UNIX sites, an alternative mail server to consider is MMDF, which is now at version 2. MMDF is distributed as part of the SCO UNIX release, and is also available in the user contributed portion of 4.3BSD. Though free, MMDF is licensed, and resale is restricted. Sites running MMDF should be on the mmdf email list; requests to join this list should be sent to: `mmdf2-request@relay.cs.net`.

Trojan Horses

Programs that masquerade as legitimate system code but which contain trap doors or other aides to unauthorized access are known as *Trojan Horses*. Computer "viruses," intrusive software that infects seemingly innocent programs and propagates when the infected programs are executed or copied, are a special case of Trojan Horse programs. (Virus attacks have been seen against PCs, but as yet have rarely been directed against UNIX systems).

To guard against Trojan Horse attacks, be wary of programs downloaded from remote sources. At minimum, do not download executables from any but the most trusted sources. Also, as noted above, to avoid proliferation of "infected" software, checksums should be computed, recorded, and periodically verified.

CERT

The Internet community can get security assistance from the *Computer Emergency Response Team* (CERT), established by DARPA in November 1988. The *Coordination Center for the CERT* (CERT/CC) is located at the Software Engineering Institute at Carnegie Mellon University. The CERT is intended to respond to computer security threats such as the November 1988 worm attack that invaded many defense and research computers. [Ed.: See separate article on the CERT on page 44 of this issue].

CERT assists Internet sites in response to security attacks or other emergency situations. It can immediately tap experts to diagnose and solve the problems, as well as establish and maintain communications with the affected computer users and with government authorities as appropriate. Specific responses will be taken in accordance with the nature of the problem and the magnitude of the threat.

CERT is also an information clearing-house for the identification and repair of security vulnerabilities, informal assessments of existing systems in the research community, improvement to emergency response capability, and both vendor and user security awareness. This security information is distributed by periodic bulletins, and is posted to the USENET news group `comp.security.announce`.

In addition, the security advisories issued by CERT, as well as other useful security-related information, are available via anonymous FTP from `cert.sei.cmu.edu`.

For immediate response to attacks or incidents, CERT mans a 24-hour hotline at 1-412-268-7090. To subscribe to CERT's security announcement bulletin, or for further information, contact:

CERT
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
1-412-268-7080 `cert@cert.sei.cmu.edu`.

SCC

For DDN users, the *Security Coordination Center* (SCC) serves a function similar to CERT. The SCC is the DDN's clearing-house for host/user security problems and fixes, and works with the DDN Network Security Officer. The SCC also distributes the DDN Security Bulletin, which communicates information on network and host security exposures, fixes, and concerns to security and management personnel at DDN facilities. It is available online, via Kermit or anonymous FTP from `nic.ddn.mil`, in `SCC:DDN-SECURITY-yy-nn.TXT` (where "`yy`" is the year and "`nn`" is the bulletin number). The SCC provides immediate assistance with DDN-related host security problems; call 1-800-235-3155 (6:00 a.m. to 5:00 p.m. Pacific Time) or send e-mail to `SCC@nic.ddn.mil`. For 24 hour coverage, call the MILNET Trouble Desk 1-800-451-7413 or AUTOVON 231-1713.

The CERT/CC and the SCC communicate on a regular basis and support each other when problems occur. These two organizations are examples of the incident response centers that are forming; each serving their own constituency or focusing on a particular area of technology.

Other network groups that discuss security issues are: `comp.protocols.tcp-ip`, `comp.virus`, `misc.security`, and the BITNET Listserv list called `VIRUS-L`.

Internet information

This tutorial is an adaptation of an appendix found in RFC 1147, "*A Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices.*" We encourage readers interested in network management to examine that catalog.

There are many available references on the TCP/IP protocol suite, the internet architecture, and the DDN Internet. An excellent place to begin is with the recent IETF document, "Where to Start: A Bibliography of General Internetworking Information." This is a list of online and hard copy documents, reference materials, and multimedia training tools that address general networking information and "how to use the Internet." It presents a representative collection of materials that will help the reader become familiar with the concepts of internetworking.

The bibliography is available by anonymous FTP from host `nic.ddn.mil`:

Directory: `internet-drafts:`
Filename: `draft-ietf-userdoc-biblio-01.txt`

(The file is also available from host `nnsc.nsf.net`)

continued on next page

Practical Intro to Network Management *(continued)*

Inquires can be sent to `user-doc@nnsc.nsf.net` or by postal mail to:

Corporation for National Research Initiatives
1895 Preston White, Suite 100
Reston, VA 22091
Attn.: IAB Secretariat.

Books

Two texts on networking are especially noteworthy. *Internetworking With TCP/IP*, by Douglas Comer, is an informative description of the TCP/IP protocol suite and its underlying architecture. The *UNIX System Administration Handbook*, by Nemeth, Snyder, and Seebass, is a "must have" for system administrators who are responsible for UNIX hosts. In addition to covering UNIX, it provides a wealth of Tutorial material on networking, the Internet, and network management.

Online documents

A great deal of information on the Internet is available online. An automated, online reference service is available from CSNET. To obtain a bibliography of their online offerings, send the email message:

```
request: info
topic: help
request: end
```

to `info-server@sh.cs.net`.

The DDN NIC also offers automated access to many NIC documents, online files, and WHOIS information via electronic mail. To use the service, send an email message with your request specified in the `Subject:` field of the message. For a sampling of the type of offerings available through this service, send the following message:

```
To: Service@nic.ddn.mil
Subject: help
Msg: <none>
```

Implementation catalog

The *DDN Protocol Implementations and Vendors Guide*, published by the DDN Network Information Center (DDN NIC), is an online reference to products and implementations associated with the DoD Defense Data Network (DDN) group of communication protocols, with emphasis on TCP/IP and OSI protocols. It contains information on protocol policy and evaluation procedures, a discussion of software and hardware implementations, and analysis tools with a focus on protocol and network analyzers. To obtain the guide, invoke FTP at your local host and connect to host `nic.ddn.mil` (Internet address 192.67.67.20). Log in using username "`anonymous`" with password "`guest`" and get the file `NETINFO:VENDORS-GUIDE.DOC`.

The guide is also available in hardcopy form. To obtain the hardcopy version, contact the DDN Network Information Center (see p. 11). For further information about the guide, or for information on how to list a product in the guide, contact the DDN NIC. [Products mentioned in the guide are not specifically endorsed or recommended by the Defense Communications Agency (DCA)].

There are many additional online sources on Internet Management. RFC 1118, "A Hitchhiker's Guide to the Internet," by Ed Krol, is a useful introduction to the Internet routing algorithms.

For more of the nitty-gritty on laying out and configuring a campus net, see Charles Hedrick's "Introduction to Administration of an Internet-based Local Network," available via anonymous FTP from `cs.rutgers.edu`.

Subdirectory: `runet`
File: `tcp-ip-admin`.

Finally, anyone responsible for systems connected to the Internet must be thoroughly versed in the Host Requirements RFCs (RFC 1122 & 1123) and "Requirements for Internet Gateways," RFC 1009.

The Final Words

Keep smiling, no matter how bad things may seem. You are the expert. They need you!

> "Network Managers are either very thin or very thick, depending on how they deal with stress." —*Terry Hardgrave*

ROBERT H. (Bob) STINE is with Applied Cybernetics, Inc. He has been analyzing and prototyping data communiction protocols and systems for six years. Bob earned his M.S. in Computer Science from George Washington University (1984), and a B.A. in History from the University of Alabama (1979, cum laude). He is currently developing a rule base to support the automated management of a major U.S. carrier's network.

J. PAUL HOLBROOK is a technical coordinator for the Computer Emergency Response Team (CERT) at Carnegie Mellon University's Software Engineering Institute. He is a co-chair of the Site Security Policy Handbook Working Group, which is part of the Internet Engineering Task Force. Prior to joining CERT, he spent seven years at Xerox Corporation working on the Xerox Star (later Viewpoint) system. He has a B.S. in computer science from the University of California at Irvine.

JAMES VanBOKKELEN's undergraduate relationship with MIT ended (scoreless tie) in 1980. From 1980 to 1985, he was Manager, Software Development for Perception Technology Corp., working on touch-tone data entry/voice response devices (does the IRS "Teletax" system ring a bell? No? Oh, well.) In 1986, he helped found FTP Software Inc., becoming the company's first VP of Marketing. Circumstances being what they are, he found himself getting sucked back into Software, replacing John Romkey as VP in 1987, and on the principle that the biggest pieces rise to the top, replaced Roxanne VanBokkelen as President in 1988. Most of his time is spent working on projects no one else here will touch, like the IETF Host Requirements Working Group, RFC 1001/1002 NetBIOS, TCP subtleties, and answering lots of e-mail.

MIKE PATTON was an undergraduate at MIT on and off for 9 years, finally working with Dave Clark in the Laboratory for Computer Science (LCS) in the early days of TCP/IP and subsequently getting his BS in 1982. Thus inspired to explore the world outside MIT, he went to work for James VanBokkelen at PTC and took over as Manager, Software Development when James left. Circumstances being what they are, he found himself getting sucked back to MIT in 1987 where he was hired by Dave Clark as Network Manager for LCS. Managing a network with over 300 researchers, most of whom want it to be a utility and a few of whom want it to be a base for experiments and all of whom are capable of modifying it to "fix" problems, has proven to meet all the criteria of "interesting" as used in the Chinese curse, but it can be fun.

A Brief History of Network Management of TCP/IP Internets

by Marshall T. Rose, PSI, Inc.

Introduction

This article briefly introduces the history and current status of network management efforts for the Internet suite of protocols (commonly known as "TCP/IP"). In particular it focuses on the technology as it is currently standardized.

Background

The unprecedented success of the Internet suite of protocols has led to the construction of large communications infrastructures, termed *internets*. These are composed of wide and local area networks and consist of *end systems*, such as hosts, terminal servers, and printers; *intermediate systems*, such as routers; and, *media devices*, such as bridges, hubs, and multiplexors. As TCP/IP is an "open" solution to computer-communications, it should not be surprising that these internets are, by their very nature, heterogeneous, multi-vendor environments.

However, the rapid growth of the number and size of internets had made network management problematic:

- Equipment additions and changes often lead to configuration errors;
- Increased scale makes former tools impractical;
- Increased heterogeneity makes proprietary tools fail; and,
- Requirements for larger staffs introduces a wider range of expertise (i.e., those with less experience), hence requiring tools which are more sophisticated, yet easier to use.

In addition to the need to keep today's networks running, there is also a need for traffic and utilization data, so as to design and plan new extensions in addition to justifying these extensions.

Conventional wisdom

Conventional wisdom dictates that a network management system contains three components:

- At least one *Network Management Station* (NMS);
- Several network elements or *managed nodes*, each containing an *agent*; and,
- A network management *protocol*, which is used by the station and the agents to communicate management information.

As might be expected, the method used to augment an element with an agent may have substantive impacts on the element's performance.

Once a common protocol and communications path exists between the station and the agent, information is exchanged using a machine-independent data encoding language. Typically two notations are employed: an *abstract syntax*, which is a description language used to define the data structures exchanged by the protocol; and, a *transfer syntax*, which defines the encoding rules for representing instances of those data structures (values) as bits in a transmission.

Finally, it should be noted that any well-designed network management system must satisfy these architectural principles:

- *Universal:* network management must be end-to-end, as internets connect a wide range of platforms across a diverse mix of media;
- *Economical:* network management must be sensitive to the impact of the agent on the various elements' platform requirements (speed and memory); and,
- *Extensible:* network management is, at present, understood only at the most superficial levels.

It should be noted that the economical behavior principle, combined with the notion that there are more agents than stations, imply that the majority of burden should be placed on the stations, not the agents (and elements).

The first round

Although several ad hoc methods for network management existed in the Internet suite of protocols, no official approach existed. So, in February of 1988, an ad hoc committee met at the request of the *Internet Activities Board* (IAB—the body which oversees the technical development of the Internet suite of protocols). The committee's charter was to examine the current state of network management technology and provide recommendations to the IAB as to what steps should be taken to develop an Internet standard for network management. After consideration of the committee's deliberations, the IAB issued RFC 1052 [7] detailing their strategy.

RFC 1052

The IAB strategy in RFC 1052 called for the development of a management framework for the Internet suite of protocols which was management protocol-independent. This direction was taken in the hope that it would ease the future transition to OSI-style network management, if OSI technology would ever mature to the point of production capability.

After setting this direction, the IAB strategy called for development of a short-term solution, based on existing technology, along with exploration of the OSI network management framework for use in TCP/IP-based internets.

A common framework

The working group charged with developing this protocol-independent framework had quite a difficult task: in addition to enumerating the objects that could be managed, these objects had to be defined in a way which was independent of the actual management protocol in use. This protocol-independence is critical if more than one management protocol is ever to be used.

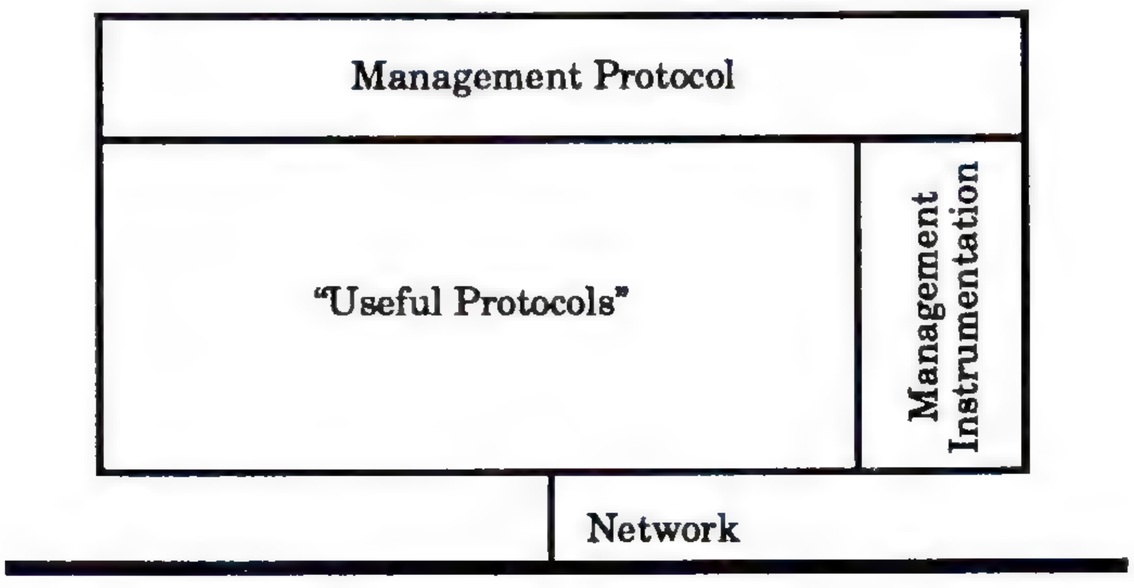

Figure 1: High-level Architecture of a Network Element

continued on next page

A Brief History of Network Management *(continued)*

A NMS usually resides on a workstation or mainframe as applications software. In contrast, an agent is usually integrated across the layers of a network element. Consider this high-level architecture of a network element as shown in Figure 1. It should be clear that a well-designed agent implementation could support multiple management protocols without a great deal of difficulty. In a transition environment this is paramount.

From Figure 1, we note that there are three parts:

- "Useful" protocols, which do "real" work;
- A *management protocol*, which permits the monitoring and control of the "useful" protocols; and,
- *Management instrumentation*, which interacts with the local implementation of the "useful" protocols in order to achieve monitoring and control.

In order to do network management, we think of the "useful" protocols as containing *managed objects*, such as routing tables, interface information, and so on. In an implementation, the management instrumentation provides these managed object abstractions to the management protocol. Thus, to the extent that these managed objects are defined in a fashion that is independent of the management protocol, the management instrumentation is also independent of the management protocol. This means that one could conceivably exchange the management protocol used on a node without changing either the "useful" protocols or the management instrumentation.

Further, in any well-designed system, the cost of implementing and maintaining the "useful" protocols should be at least an order of magnitude greater than the associated costs of the management instrumentation and management protocols. Thus, if the objects are defined in a management protocol-independent fashion, a significant savings may be realized.

Another benefit of the common framework is obtained in management stations. If a transition occurs, there will undoubtedly be a mixture of managed nodes using one or more of the management protocols. By having a common framework, the network operator can be presented with a common interface for managing the network, independent of what actual protocols are used to manage individual nodes.

To achieve this independence, the working group divided its task into two parts. The first part was to develop a *Structure of Management Information* or SMI. The SMI defines the rules for how managed objects are described and how management protocols may access these objects. The second part was to develop a *Management Information Base* or MIB. This is the collection of defined objects that can be accessed via network management protocol.

SMI

The SMI is defined in RFC 1065 [8] and is largely an administrative document: it says "how to do things," but not actually "what to do." It is the MIB and the management protocol, which are responsible for the "what to do" part, and the SMI is used strictly to provide a well-defined interface between them.

The SMI philosophy is to foster:

- *Simplicity*: because general understanding of network management is (at this time) very limited; by taking a simple approach, future extensions will be less constrained. In addition, there is hope that today's systems will be more workable.

- *Extensibility*: because there are many possible approaches which might be followed; by emphasizing extensibility, a larger number of future approaches might be attempted. In addition, since today's systems will be around for some time, it is essential to provide a straight-forward way for the new to work with the old.

ASN.1

The *object information model* used in the SMI is simple. Managed objects are defined using a data description language called ASN.1 (the OSI *Abstract Syntax Notation One*). ASN.1 is useful for describing information object structures in a machine-independent fashion. In addition, ASN.1 definitions can be written so they convey to a human reader the semantics of the objects they define.

The SMI defines an ASN.1 macro, `OBJECT-TYPE`, that the MIB uses when managed objects are defined. Here's an example:

```
sysUpTime OBJECT-TYPE
     SYNTAX TimeTicks
     ACCESS read-only
     STATUS mandatory
     ::= { system 3 }
```

This simple definition captures the following semantics:

- A managed object called `sysUpTime` is defined;
- This object contains `TimeTicks` information (this data type is also defined in the SMI—it is a measurement of time in hundredths of a second);
- This object is read-only, so that management protocols may not attempt to modify this object;
- This object is mandatory, so that all managed nodes (gateways, hosts, etc.) must provide this object; and,
- When a management protocol accesses this object, it uses the name `{ system 3 }`

This is all intuitive except for the last part.

Object Identifiers

All managed objects must have a name which can be used by a management protocol when it wishes to identify that object. The SMI uses the ASN.1 OBJECT IDENTIFIER notion for this purpose. An OBJECT IDENTIFIER is an administratively assigned name. The top-level numbers are allocated by the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC).

In turn, the ISO/IEC has allocated some of the numbers to various national standards bodies. In turn, one of these bodies allocated a number to the U.S. Department of Defense. Finally, one of these numbers, 1.3.6.1.2, was allocated for TCP/IP-based network management.

continued on next page

A Brief History of Network Management *(continued)*

The SMI takes it from here, by saying how numbers that are subordinate to 1.3.6.1.2 are used. The name `{ system 3 }` is such a number which identifies the object as occurring in the current Internet MIB, in the system group, as the third defined object.

In addition to the Internet standard MIB, the SMI also allocates name-space for experimental objects and enterprise-specific objects. For the latter, vendors and organizations can apply for their own number and receive the delegated authority for defining their own managed objects.

One of the "simple" aspects of the SMI was to restrict the kind of data that could be used in a managed object. Many powerful and complex structures can be defined using ASN.1. These structures are slow to encode when transmitted on the network; they are also hard to decode when received from the network. In order to make network management available to the largest range of TCP/IP-based products, it was decided that the ASN.1 used to represent the "syntax" of a managed object should be limited. The SMI defines the rules which mandate the limitations.

The SMI does not define access mechanisms used by the management protocols. For example, the SMI permits aggregate types, such as tables, to be defined, but does not define the rules used for table access. These are mechanisms specific to the management protocol.

MIB The MIB is defined in RFC 1066 [5] The Management Information Base is largely a technical document: for each layer in the Internet suite of protocols, the MIB defines the objects which may be managed. The first version of the MIB, sometimes referred to as MIB-I, was defined to capture the basic essential aspects of the fundamental protocols. It is intended that each new version will add objects, but will change only a few, as they "fill in the gaps."

MIB-I itself is divided into several groups of variables:

group	no.	objects for
system	3	the managed node itself
interfaces	22	network attachments
at	3	IP address translation
ip	33	the Internet Protocol
icmp	26	the Internet Control Message Protocol
tcp	17	the Transmission Control Protocol
udp	4	the User Datagram Protocol
egp	6	the Exterior Gateway Protocol, used by nodes directly attached to the Internet backbone
total	114	

Each managed node does not have to support all of these groups. Each node only needs to support only those groups which are appropriate (e.g., IP gateways needn't support the TCP group). However, if a group is appropriate, then all objects in that group must be supported.

There were extensive criteria for deciding if an object should be defined in MIB-I. Perhaps the two most important criteria were that:

- An object be needed to configure or diagnose a node; or
- An object was previously known to be useful for management.

Further, the working group had as a goal to produce a first version of the MIB with no more than 100 objects. If this could be accomplished, then it would be easier for vendors to fully implement MIB-I. (When the MIB-I definition was completed, 114 objects were defined.)

SNMP

As noted earlier, the short-term network management protocol approved by the IAB was to be based on existing Internet technology. Thus, a working group was formed to modify the existing *Simple Gateway Monitoring Protocol* (SGMP) [3], to reflect the experience gained from its use in operational networks, and then align it with the SMI/MIB. The resulting protocol was called the *Simple Network Management Protocol* (SNMP), which is defined in RFC 1098 [2].

SNMP is a request/response protocol with very few operators:

- *get:* to retrieve instances of objects;
- *get-next:* to traverse the tree of objects; and,
- *set:* to modify instances of objects.

In addition SNMP supports a trap-directed polling model, so there is also a *trap* operator.

SNMP is normally layered on top of the *User Datagram Protocol* (UDP), which offers a connectionless-mode transport service in the Internet suite of protocols. Each SNMP message consists of:

- A *version number,* for extensibility purposes;
- A *community string,* used to define the authentication *policy* for the message; and,
- A *protocol data unit,* containing the operator, request identifier, and associated operands.

[Note the use of the term "policy." Any semantics associated with the policy, are defined entirely via bilateral agreement. For example, a management station might wish to encrypt the protocol data unit associated with a message (for privacy) or to attach an integrity checksum (for authentication). The community name implicitly identifies these policies and defines the mechanisms used to achieve these goals.]

It should be noted that the transport requirements of SNMP are purposefully conservative: because of its unique properties, network management is unlike any other application running in the network. As such, it was felt that transport mechanisms developed for typical end-to-end service are inappropriate. For example, concerns for robustness and survivability argue that the NMS, and not the transport service, should control the level of retransmission.

For a more thorough explanation of the elegance of the SNMP philosophy, the reader should consult [1].

continued on next page

A Brief History of Network Management *(continued)*

Where we are now

In August of 1988, a scant six months after the first meeting of the IAB's ad hoc committee, the SMI, MIB-I, and SNMP specifications were completed, independently implemented, and declared to be draft standards by the IAB. Further, several implementations of these documents appeared in vendor products, by the end of August, 1988.

In April of 1989, SNMP was elevated to "Recommended" status, and became the *de facto* operational standard for network management of TCP/IP-based Internets, as evidenced by widespread vendor support and the number of deployed systems. For example, more than thirty vendors demonstrated SNMP products at the INTEROP® trade-show in October of 1989.

Unfortunately, work on the use of the OSI management framework had failed to produce any workable TCP/IP implementations in the same time-frame. This left SNMP as the only workable choice for the management of multi-vendor internets.

The second round

After the first round of network management technology, which produced SNMP, had stabilized, work began on the first set of revisions to the Internet-standard MIB. Unfortunately, the SNMP and OSI camps could not reach agreement on this work. Another meeting of the IAB's ad hoc committee was called, with the outcome reported in RFC 1109 [7] that the two camps would be allowed to define divergent SMI and MIB specifications, and that the working group that had produced the common framework was to be disbanded!

It is beyond the scope of this article to comment on the political ramifications of this decision, other than to note that:

- From one perspective, the decision spelled the doom of OSI network management in the Internet, as SNMP was providing useful service, and the OSI approach had no credibility (i.e., no working implementations); and,
- From the opposite perspective, the decision gave the OSI proponents the opportunity to exploit the richer functionality of the OSI management framework.

In May of 1990 the IAB elevated the SMI, MIB and SNMP to "Standard Protocols" with "Recommended" status, and re-issued these as RFCs 1155, 1156 and 1157.

MIB–II

Regardless of the "spin" placed on this decision, after several months' experience, it was clear that work on MIB-II had to be started. To achieve this, the working group that created SNMP was re-constituted to produce MIB-II. The approach taken was two-fold:

- First, MIB-II would reflect new operational requirements; and,
- Second, heavy use would be made of the experimental MIB space, so as to "prove" new objects before placing them in the Internet-standard space in the MIB.

To achieve the first goal, the SNMP Working Group continued in the tried-and-true Internet tradition:

- A draft was prepared,
- The group examined this draft and reached tentative agreement,

- Members of the group independently implemented the draft and reported back their experiences, and,
- When no significant problems were found, the draft was ratified.

Work began on the draft in August of 1989 and consensus was reached in December, 1989. In June of 1990, the IAB elevated MIB-II to a "Proposed Standard," and re-issued it as RFC 1158.

Features of MIB-II

The overriding goal of the MIB-II work was to maintain compatibility with the Internet-standard SMI and MIB-I and SNMP. Keeping in mind this requirement, there were three areas to be addressed:

- Incremental additions to reflect new operational requirements;
- Improved support for multi-protocol engines; and,
- Textual clean-up to improve clarity.

The most interesting textual clean-up was the introduction of a new data type, `DisplayString`. This is used to distinguish between data which is binary in nature (such as MAC addresses), and data which is textual in nature (from the Internet *Network Virtual Terminal* (NVT) ASCII repertoire, as defined in [4].

To briefly compare MIB-II to its predecessor:

group	no.	comments
system	7	was 3
interfaces	23	was 22
at	3	will be 0
ip	38	was 33
icmp	26	unchanged
tcp	19	was 17
udp	7	new table
egp	18	expanded table
transmission	0	new
snmp	30	new
total	171	50% larger

There are three areas of interest:

- *address translation:* MIB-I provided for one-way mapping of protocol addresses to physical addresses, but components of other network protocols (e.g., ES-IS) require inverse mappings. Unfortunately, indexing a single table to provide mappings in both directions is too difficult for some implementations. As such, the address translation table was *deprecated* (marked as being eventually obsolete), and each protocol family now introduces one or two tables for mappings in the appropriate direction.
- *transmission:* MIB-II defines a new group containing objects for each type of interface (Token Ring, loopback, etc.). These definitions are introduced in the experimental or vendor space of the MIB and eventually transfer to the Internet-standard space when consensus is reached.
- *snmp:* MIB-II defines a new group containing objects about SNMP. This allows the NMS to manage the SNMP portion of the entities that it manages.

continued on next page

A Brief History of Network Management *(continued)*

The future

The future is hard to predict, but there are three areas in which the trends seem pretty obvious to the author:

For the MIB: Many more objects will be defined in the experimental space. As these are proven and consensus is reached, future versions of the Internet-standard MIB will incorporate these objects.

On the short-term horizon, the SNMP Working Group has developed a set of definitions for the OSI connectionless-mode network protocol (CLNP) objects, similar to the IP portion of MIB-II. The idea here is to leverage the existing tools that are capable of managing IP-based networks.

For SNMP: Full deployment of SNMP in TCP/IP-based internets seems imminent given the high-level of success it is enjoying. In addition to more than thirty vendors supplying SNMP as a part of their networking product line, there are now three reference implementations which are openly available. One of these will be included with the next release of the popular Berkeley Standard Distribution (BSD) of UNIX.

In addition, the SNMP Working Group has completed a document describing how SNMP can be mapped onto the OSI transport service (both connection-oriented and connectionless-mode), and a reference implementation of this technology has already been produced. Thus, whilst SNMP may not be an OSI standard, it may very well soon become the de facto protocol of choice for managing OSI networks.

In general: The most pressing need is currently that of authentication. Although SNMP provides the hooks for authentication, it (wisely) does not define the mechanisms. Fortunately, a working group is rapidly progressing towards producing a draft which allows for both authentication and privacy of SNMP messages.

References

[1] Jeffrey D. Case, James R. Davin, Mark S. Fedor, & Martin L. Schoffstall, "Network Management and the Design of SNMP," *ConneXions*, Volume 3, No. 3, March, 1989.

[2] Jeffrey D. Case, Mark S. Fedor, Martin L. Schoffstall, & James R. Davin, "A Simple Network Management Protocol," RFC 1098, April 1989.

[3] James R. Davin, Jeffrey D. Case, Mark S. Fedor, & Martin L. Schoffstall, "A Simple Gateway Monitoring Protocol," RFC 1028, November 1987.

[4] Jonathan B. Postel, "TELNET Protocol Specification," RFC 854, May 1983. (See also MILSTD-1782).

[5] McCloghrie, K. & Rose, M. T., "Management Information Base Network Management for TCP/IP based internets, RFC 1066, August 1988.

[6] Vinton G. Cerf, "IAB Recommendations for the Development of Internet Network Management Standards," RFC 1052, April 1988.

[7] Vinton G. Cerf, "Report of the Second Ad Hoc Network Management Review Group," RFC 1109, August 1989.

[8] McCloghrie, K. & Rose, M. T., "Structure and identification of management information for TCP/IP-based internets," RFC 1065.

This work was partially supported by the U.S. Defense Advanced Research Projects Agency (DARPA) and the Rome Air Development Center of the U.S. Air Force Systems Command under contract number F30602–88–C–0016. The content of the information contained herein does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred.

MARSHALL T. ROSE is Principal Scientist at Performance Systems International, Inc., where he works on OSI protocols and network management. He is the principal implementor of the *ISO Development Environment* (ISODE), an openly available implementation of the upper layers of the OSI protocol suite. He is the author of *The Open Book: A Practical Perspective on OSI*, and *The Simple Book: An Introduction to Management of TCP/IP-based internets*, both professional texts published by Prentice-Hall. Rose received the Ph.D. degree in Information and Computer Science from the University of California, Irvine, in 1984. By international agreement, he no longer allows the publication of self-aggrandizing biographies, and hopes others in industry will follow his example in this matter. His subscriptions to *The Atlantic* and *Rolling Stone Magazine* are in good standing.

IETF Working Groups on Security

The *Internet Engineering Task Force* (IETF) has several efforts underway to address security for the Internet. At the time of this writing, the following groups exist. If you want to join their electronic discussion, be sure to use the "-request" convention, e.g., send your request to be added to the list to `spwg-request@nri.reston.va.us` rather than to the main list.

The IETF *Security Policy Working Group* (SPWG) is chartered to create a proposed Internet Security Policy for review, possible modification, and possible adoption by the Internet Activities Board. The SPWG will focus on both technical and administrative issues related to security, including integrity, authentication and confidentiality controls, and administration of hosts and networks. The mailing list for this activity is: `spwg@nri.reston.va.us`

The IETF *Site Security Policy Handbook Working Group* (SSPHWG) is chartered to create a handbook that will help sites develop their own site-specific policies and procedures to deal with computer security problems and their prevention. The mailing list for this group is: `ssphwg@cert.sei.cmu.edu`.

The IETF *Authentication Working Group* (AWG) discusses issues relating to providing for the security and integrity of information on the Internet, with emphasis on those protocols used to operate and control the network (e.g, IP and SNMP). The group will propose open standard solutions to problems in network authentication. The mailing list for this activity is: `awg@bitsy.mit.edu`.

Ed.: Network Management and Network Security are both covered extensively at INTEROP 90.

Case Study: Using SNMP to manage a large network

by Mark S. Fedor, PSI, Inc.

Introduction

The protocol has been specified; the prototypes developed; and the documents written. What next? The real test of a protocol specification is how it performs in the real-life world of internetworking. How easy is it to implement? How easy can the protocol features be taken advantage of? Does it solve the problems it was intended to solve? These questions and many more determine the life path and evolution of an internet protocol.

All too often, papers and articles on network management concentrate on protocol internals or comparisons of one network management protocol to another. What is often overlooked is the practical experiences gained through the use of network management. What network management tools are needed? Which tools are used the most? How do you go about monitoring the network? This article will convey some of the experiences gained in using SNMP to manage PSINet.

Background

PSINet is a TCP/IP-based network which has a backbone that spans Boston to LA. There are currently 62 sites connected to PSINet as well as a number of terminal servers dedicated to individual dial-up and UUCP. PSI, Inc. manages PSINet using SNMP software which was developed originally by the Research and Development team at NYSERNet, Inc. The software has continued to be developed and maintained by the Research and Development group at PSI, Inc.

PSI uses equipment from diverse vendors to implement the network and provide network services. Because of the multi-vendor, multi-service environment, managing PSINet poses a significant challenge.

Managing a large wide area network

In the real-world situation of managing a large network, there are a number of goals which a network manager must strive for. First and foremost is to simply keep the network running. The network manager must respond quickly and efficiently to arbitrarily complex problems. Trends must be analyzed and watched so intervention can take place before a problem starts. This is called being *proactive* as opposed to being *reactive*. It is critical to be one step ahead of the network users; the network manager should not first hear about a problem by being notified by a network user.

The network manager must deal with increasing growth. He/she must know when to engineer more bandwidth and processing cycles into the network. In order to do this, network information must be collected and made understandable.

The task of a network manager is a difficult one. The management protocol must assist him in performing the tasks outlined above. This is accomplished by the *Network Management Station* (NMS) interactively reporting on link status, routing, general errors, and certain network events. Information logging and error logging is another tool which the network manager can use to perform these tasks.

The PSINet NOC

The PSINet network operations staff consists of 6 people. The *Network Operations Center* (NOC) is located in Troy, NY and manned 24 hours a day, 7 days a week. The primary NMS is a Sun 3/60 with a color bitmap display. There are various other X Terminals and ASCII terminals around the NOC for the operators to perform everyday tasks like mail reading and file editing. The software running on the primary NMS is the PSINet SNMP distribution. The windowing system is X version 11 release 3 software from MIT.

Real-time tools

The primary function of the color display is to provide real-time monitoring of PSINet. At all times, a SNMP monitoring tool, *snmpxmon*, displays a map of PSINet and the state of its links and nodes (see Figure 1). In this application, a red line means the link is down, green means the link is up, and yellow means the link is in an unknown state. The application polls all PSINet nodes at 30 second intervals asking for link state information via SNMP.

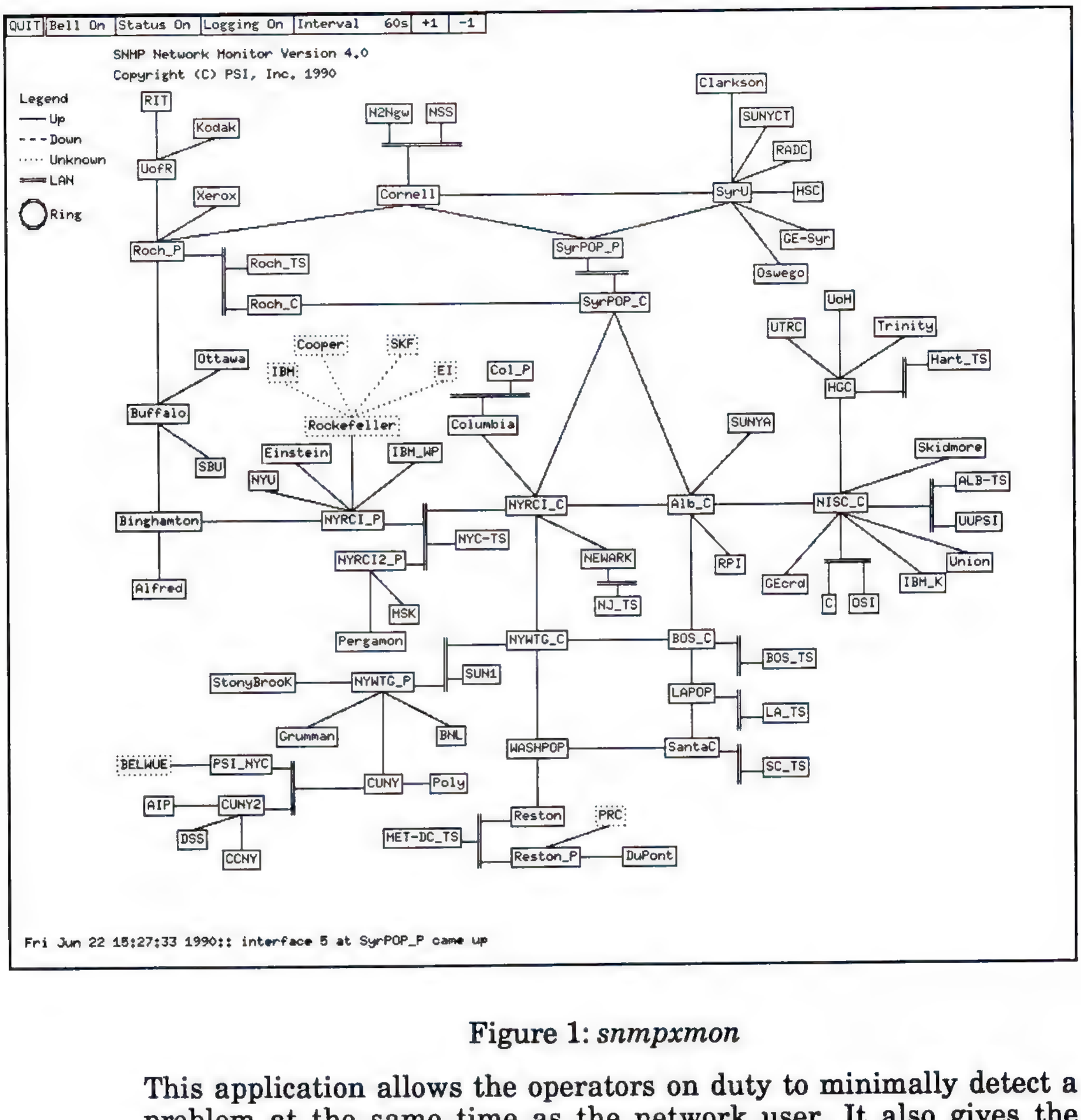

Figure 1: *snmpxmon*

This application allows the operators on duty to minimally detect a problem at the same time as the network user. It also gives the operator a global picture of the entire state of the network at any given time. This knowledge is critical in having the operator be proactive instead of reactive. For instance, if the operator notices a link fluctuating from green to red and back to green continuously, the operator can launch a more detailed investigation into the link in question with other SNMP tools and most likely set a solution into motion before anyone notices a network problem.

Another application, which continuously runs on the color display, provides real-time monitoring of the state of routing within PSINet. This application, called *snmpxrtmetric,* retrieves information from the routing tables of the routers and presents the routing metrics to specified destination networks in a bar graph format (see Figure 2 on the following page).

continued on next page

Using SNMP to manage a large network *(continued)*

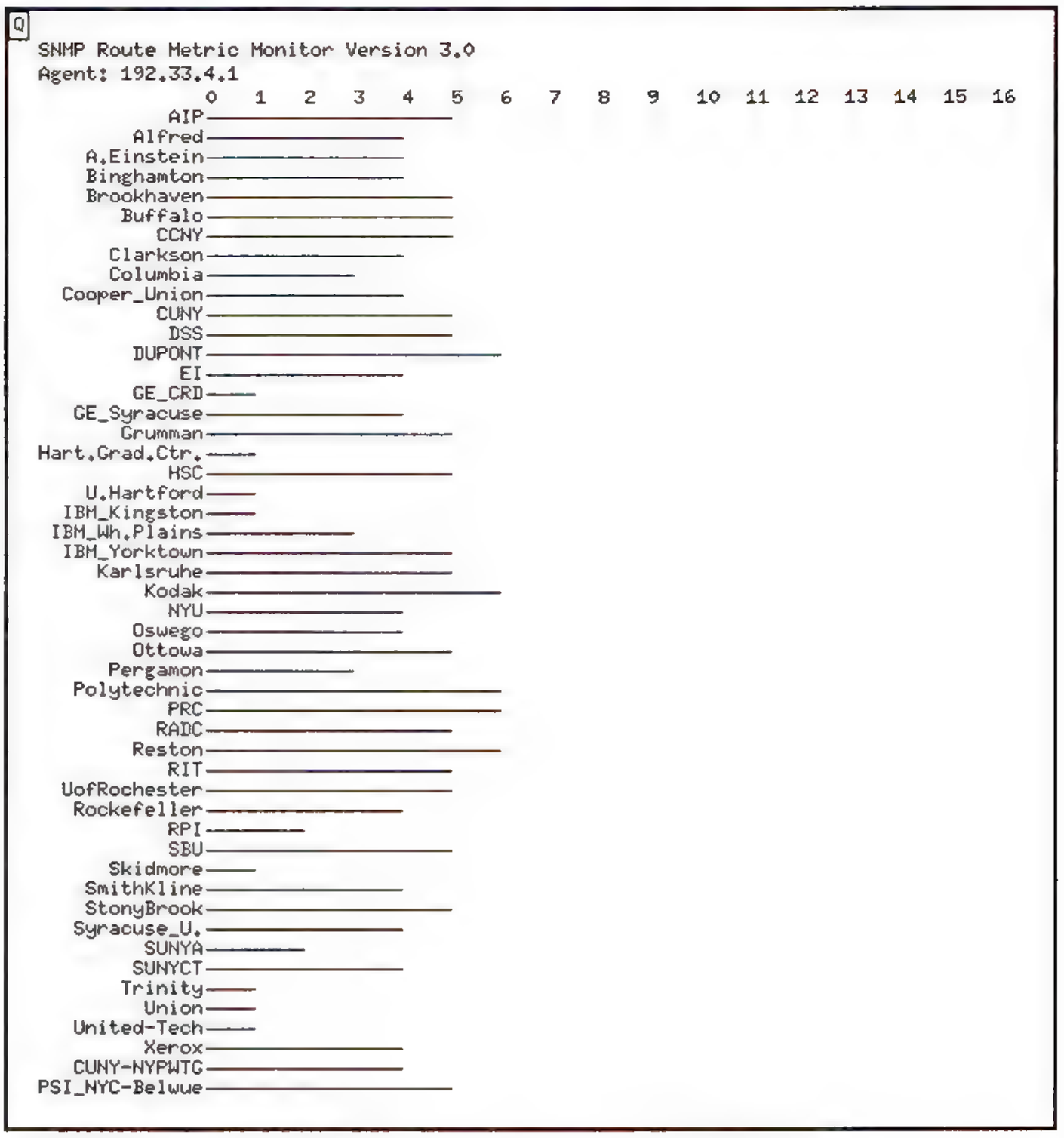

Figure 2: *snmpxrtmetric*

Every destination network has an ideal routing metric or a "best" routing metric. If, on the *snmpxrtmetric* display, an ideal metric to a destination network suddenly increased by four hops, this would give an indication to the network operator of a change in state somewhere in the network. Coupled with the *snmpxmon* display, the network operator can tell whether it is a line failure, node failure, or an external event which caused the state change.

Experience has shown that the *snmpxmon* map display coupled with the route metric monitor provide sufficient indication of network problems and also provide enough information for the network operator to pinpoint further investigative efforts.

Other real-time monitoring tools which are used to give the operator a more focused or detailed view include *snmpxperf, snmpxbar,* and *snmpxperfmon. Snmpxperf* charts a single MIB variable in an EKG-style graph. *Snmpxbar* charts a single MIB variable in a bar graph and reports when a threshold has been reached. *Snmpxperfmon* charts statistics on all interfaces of a gateway using multiple EKG-style graphs.

These graphical tools are not displayed on the NMS display all the time. They are used when more information is needed from a particular network entity. For example, the operator notices a network link which is fluctuating up and down. The operator might want to look at a graph which plots the current errors on the link. The *snmpxperf* application could then be invoked and temporarily displayed on the NMS display.

A *curses* based terminal server monitoring application is used to continuously monitor the cisco Systems terminal servers installed throughout PSINet. *Snmptswho* shows terminal server activity and has the ability to log terminal server sessions into a flat file. This is very useful in determining usage patterns on the terminal servers and allows the PSI engineering staff to expand the services when and where they are needed.

Other tools

Interactive, ASCII-based firefighting tools play an important role in managing PSINet. While the real-time monitoring tools give the network operator a good head start on tracking down a problem, it is up to him to interpret the real-time graphical data and plan a course of action. This course of action would entail interactive querying, via SNMP, of the problem network devices. The querying would provide more detailed information than the graphical display so the network operator could diagnose and correct the network problem.

SNMP tools which are in use at PSINet and fill this role include *snmpsrc, snmplookup, snmproute,* and *snmpwatch. Snmpsrc* traces and reports back the path that data is taking from a starting location to a given destination network. This tool is invaluable in finding routing loops and routing black holes. It is also a good tool to use to find out how the network routing algorithm is handling the network topology. *Snmpsrc* is the most widely used interactive tool by the PSINet operations staff. *Snmplookup* allows the network operator to conduct a one-on-one SNMP session with a network entity. The network operator may ask any detailed SNMP question of the network entity and will be provided with the answer. *Snmproute* provides a way for the network operator to interrogate a network router for its routing entries. The *snmpwatch* application monitors variables and reports when the values of the variables have changed.

All of the tools mentioned were developed to fill a network management need. The development team and operations staff of PSINet have worked together to design tools which have been useful in running PSINet and other computer networks. The network management tools described above were also not meant to totally replace existing network trouble-shooting tools. The network ninja would surely use these SNMP tools to complement existing tools such as *ping* and *traceroute.*

Trend Analysis and Report Generation

Report Generation is often overlooked in the total network management scheme. Data collection is very important for a variety of reasons. First, network usage patterns can be analyzed to provide the network managers with ideas on future network planning and enhancement. Second, data collected can be analyzed for the identification of potential problems before they become acute. Third, chronic problems can be detected. Lastly, the data collected can be used for accounting and accountability.

The amount of data collected by automated tools will usually overwhelm the network manager. There is a need to postprocess the raw data collected by such tools to convert them into reports that convey useful information. Figure 3 shows the report generation process.

Within PSINet, the SNMP data collection tool, *snmppoll,* collects data from the network every fifteen minutes. The primary data collector is a Sun 3/280 and resides at the PSINet NOC. The backup data collector is a Sun 3/260 and is situated in Ithaca, NY. It is desirable to have two separate locations gathering data.

continued on next page

Using SNMP to manage a large network *(continued)*

If the primary data collector is disconnected from the network, the backup data collector will still be able to gather data from the network. The data can then be merged onto the primary data collector when it becomes time to process reports.

Every fifteen minutes, *snmppoll* polls 81 network entities on PSINet for data associated with each interface. Approximately 12 MIB variables are retrieved for each of the 237 interfaces in PSINet for a total of 2853 MIB variables retrieved each polling cycle. Approximately 23 CPU minutes and 16 megabytes of disk space are used daily.

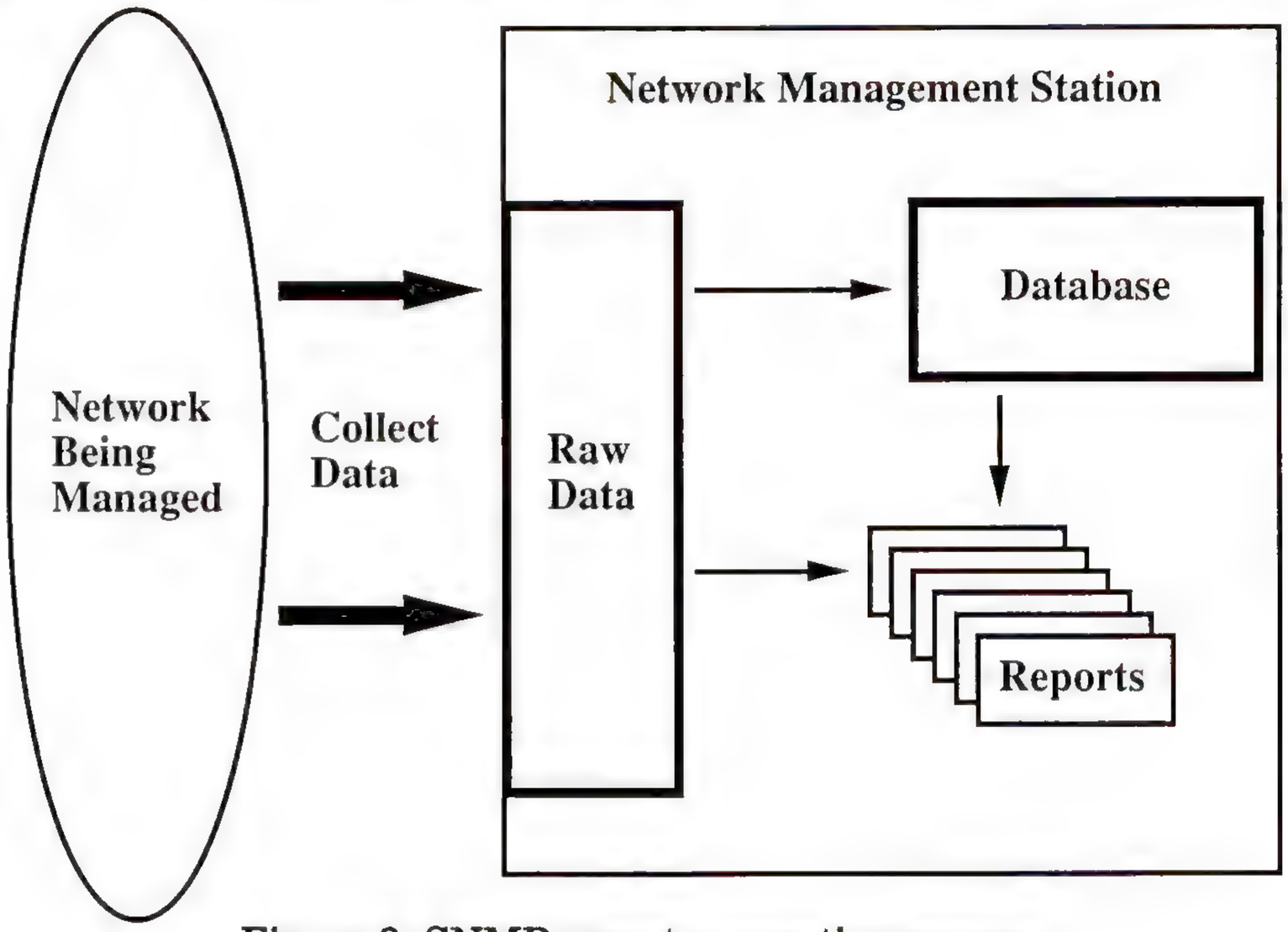

Figure 3: SNMP report generation process

At PSINet, reports from the raw data are generated in the early morning of every day. They are available for review by the operations staff when the day shift begins. The reports are used to diagnose problems and review prior events. The reports are also used to focus on operational maintenance work.

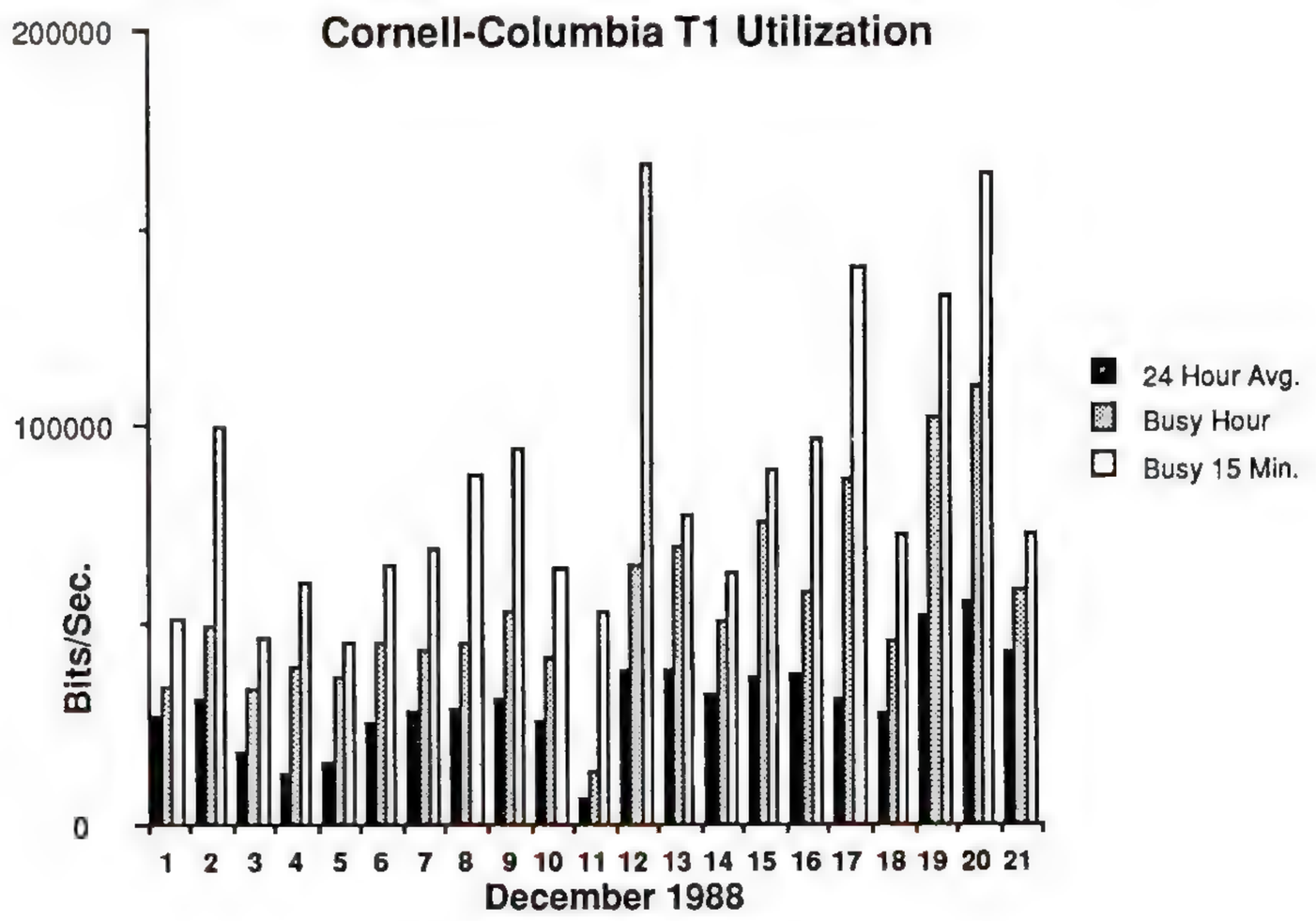

Figure 4: Link utilization

The PSINet operations staff uses eight reports each with a different level of granularity. If a problem is seen in a high-level summary report, further probing into other reports can pinpoint the problem. For instance, a high CRC error rate is seen for the day on one network link. By looking at further reports, it can be determined if the high error rate was caused by a large spike or if the error rate was caused by a consistent error count throughout the day. The PSINet reports are available via the Internet from host `nisc.nyser.net` (IP address 192.33.4.10) by anonymous FTP. The reports are located in the "`stats/`" directory. Questions can be directed to `info@psi.com`.

Graphical reports can also be generated from the raw data. Figure 4 shows a bar graph depicting the utilization of the link between Cornell and Columbia. These reports are very useful for showing network statistics to network users and providing information through presentations and proposals.

Needs of the future

While SNMP and the tools from the PSI distribution have met most of PSINet's needs of today, certain issues still need to be addressed. Experience gathered internally and information from external sources show that more work is needed in developing a user-interface which facilitates the easy use and operation of network management tools. Ideally, a complete network management system with a consistent and friendly user-interface would allow a new network operator to come up to speed faster and be more effective sooner. Look for more changes and improvements in this area as more experience is gained and network managers find out what they really want in a network management system.

References

[1] "A UNIX Implementation of the Simple Network Management Protocol," W. Yeong, M. L. Schoffstall, M. S. Fedor, *Proceedings of the 1989 Winter USENIX Conference,* San Diego, California.

[2] "Keeping it Simple,"J. Case, J. Davin, M. Fedor, M. Schoffstall, *UNIX Review,* March 1990.

[3] "Cutting Network Management Tasks Down to Size," M. S. Fedor M. S. Richards, M. L. Schoffstall, *LAN Technology,* March 1990.

[4] "Network Management and the Design of SNMP," J. D. Case, J. R. Davin, M. S. Fedor, M. L. Schoffstall, *ConneXions—The Interoperability Report,* Volume 3, No. 3, March 1989.

[5] Rose, M. T., McCloghrie, K., "Structure and identification of management information for TCP/IP-based internets," RFC 1155.

[6] McCloghrie, K., Rose, M. T., "Management Information Base for network management of TCP/IP-based internets," RFC 1156.

[7] Case, J. D., Fedor, M., Schoffstall, M. L., Davin, C., "Simple Network Management Protocol (SNMP)," RFC 1157.

[8] Rose, M. T. (Ed.), "Management Information Base for network management of TCP/IP-based internets: MIB-II," RFC 1158.

MARK S. FEDOR is a Senior Network Engineer at Performance Systems International, Inc. in Albany, New York. Prior to joining PSI, Mark was a Project Leader in the Research and Development group of NYSERNet Inc. where he was involved in the technical aspects of the New York State Education and Research Network as well as developing network management software. He also worked at the Cornell University Theory Center where he took part in the development and operation of the experimental NSFNET backbone. While at Cornell, he was the primary developer of the *gated* routing daemon for UNIX. He received a B.A. in Computer Science from State University of New York, College at Oswego in May, 1986.

Components of OSI: The Security Architecture

by James M. Galvin, Trusted Information Systems

Introduction

The *OSI Security Architecture* [1] is the most well-known of the OSI security standards. However, for completeness, we note the following additional security standards and standards with security related services exist:

Message Handling Systems [2]

Directory Authentication [3]

Data Integrity Mechanism using a Cryptographic Check [4]

ACSE Authentication [5]

...plus several in the area of Banking and Related Financial Services. [Ed.: A discussion of [2] and [3] have appeared in previous issues of *ConneXions*.]

The Security Architecture Standard defines the general security related architectural elements which can be applied when protection of communication between open systems is required or desired. The basic security services and mechanisms and their appropriate placement are identified for all layers of the *Basic Reference Model* [7]. It is important to note that while security measures in end systems, installations and organizations are important, they are only relevant to this standard when there are implications on the choice and position of security services on a communication path.

Although many more definitions and mechanisms are described by the standard, we only consider a few of the more commonly known security services here:

- integrity
- authentication
- confidentiality

Integrity

Integrity, or more precisely *data integrity*, is the property that data has not been altered or destroyed in an unauthorized manner. In this context,"altered or destroyed" is used in its most general sense. The standard identifies 5 forms of the data integrity service:

1. *Connection integrity with recovery:* provides for the integrity of all data, detecting any modification, insertion, deletion or replay of any data, with recovery attempted.
2. *Connection integrity without recovery:* as with connection integrity with recovery, but with no recovery attempted.
3. *Selective field connection integrity:* provides for the integrity of selected fields, detecting any modification, insertion, deletion or replay of said fields.
4. *Connectionless integrity:* provides for the integrity of a single connectionless data unit, detecting any modification of the data. A limited form of replay detection may be provided.
5. *Selective field connectionless integrity:* provides for the integrity of selected fields in a single connectionless data unit, detecting any modification of said fields.

There are two aspects to data integrity: the integrity of a *single data unit* or *field*; and the integrity of a *stream of data units* or fields. The first aspect deals explicitly with the modification of a data unit. A sending entity appends to the data unit a quantity which is a function of the data itself. The quantity may be a block check code, e.g., a hash calculation, or a cryptographic checkvalue, e.g., a digital signature. The receiving entity generates a corresponding quantity, which it compares to the received quantity. This is the typical situation for the connectionless services.

The connection services may also detect the insertion, deletion or replay of a data unit, the second aspect of data integrity. Protecting the sequence of data units requires some form of explicit ordering, for example sequence numbers, time stamping or cryptographic chaining. For replay protection in connectionless data transfers, the standard suggests the use of timestamping to provide limited protection.

In both aspects, it may be necessary to protect the quantity appended to the data unit. The standard recommends an encipherment mechanism but provides no explicit guidance as to choice or relevant issues.

Authentication

The standard defines two types of authentication:

1. *Data origin authentication:* corroboration that the source of data received is as claimed.

2. *Peer entity authentication:* corroboration that a peer entity in an association is the one claimed.

Similarly, the standard defines two types of authentication services:

1. *Data origin authentication:* provides for the corroboration of the source of a data unit. It does not provide protection against the duplication or modification of data units.

2. *Peer entity authentication:* provides for the corroboration of the identity of a peer entity in an association. It is provided for use at the establishment of, or at times during, the data transfer phase of a connection. It provides confidence, at the time of usage only, that an entity is not attempting a masquerade or an unauthorized replay of a previous connection.

Problems with the standard

The standard is sub-optimal with regard to its description of the mechanisms that support these services. It identifies three techniques which may be applied to authentication exchanges:

1. Use of authentication services, e.g., passwords supplied by a sending entity and checked by the receiving entity;

2. Cryptographic techniques; and

3. Use of characteristics and/or possessions of the entity.

However, no guidance is given as to how to choose an appropriate technique. In fact, except for 1 above, no specific examples are provided. Annex A provides another level detail about technique 2, but technique 3 is left obscured. Further, no discussion of the useful interrelationship of integrity and authentication exists. However, the Authentication Framework [6] is much more complete in its treatment of authentication.

continued on next page

The OSI Security Architecture *(continued)*

Confidentiality

Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities or processes. The standard identifies 4 forms of the confidentiality service:

1. *Connection confidentiality:* provides for the confidentiality of all data.
2. *Connectionless confidentiality:* provides for the confidentiality of a single connectionless data unit.
3. *Selective field confidentiality:* provides for the confidentiality of selected fields within the data on a connection or in a single connectionless data unit.
4. *Traffic flow confidentiality:* provides for the protection of the information which might be derived from observation of traffic flows.

Encipherment

Encipherment, the cryptographic transformation of data to produce ciphertext, is the only mechanism suitable to support all forms of confidentiality. Although some traffic flow confidentiality may be provided by traffic padding, this mechanism can only be effective if the traffic padding is protected by encipherment.

The existence and use of encipherment in support of confidentiality implies the use of a *key management mechanism*. Key management may involve generating suitable keys, determining who has access to keys and making available or distributing keys in a secure manner. The exchange of working keys for use during an association is a normal protocol layer function, and may or may not involve key management protocols.

The use of encipherment may involve interaction with a key manager, establishment of cryptographic parameters or cryptographic synchronization.

Summary

The standard is quite complete in both its breadth and depth of security related definitions. It could give better treatment to the relationships possible between the various services, for example integrity and authentication, but implementors are likely to notice the relationships relatively quickly.

The standard suggests that all security services be provided at the application layer of the Basic Reference Model, including all of those described here. A hint of the justification for the placement of the services at other layers is given in Annex B. Unfortunately, hints like "not considered useful at this layer" and "not appropriate at this layer" provide very little insight. For the services described above, the standard recommends the following additional placement of the service.

Both selective field forms of integrity should only be available at the application layer. However, they may be supported by encipherment mechanisms in layer 6 in conjunction with invocation and checking mechanisms in layer 7. The other 3 forms of integrity should also be available at layer 4, since this provides the true end-to-end transport connection.

Only the connection integrity without recovery and connectionless integrity should also be available at layer 3. In the case of the former, error recovery is not universally available at layer 3, so only connection integrity without recovery may be provided. Connectionless integrity is treated the same as connection integrity without recovery, in order to minimize the duplication of functions.

Both types of authentication should be provided at both layers 3 and 4, in addition to layer 7. As with integrity, the service at layer 7 may be supported by encipherment mechanisms in layer 6. In the case of data origin authentication, if peer entity authentication is provided at connection establishment time together with encipherment-based continuous authentication during the life of the connection, data origin authentication is implicitly provided. Even in the absence of peer origin authentication, data origin authentication can be provided with very little additional overhead to the data integrity service.

Connection confidentiality is recommended at all layers except layer 5, since at layer 5 it provides no additional benefit over confidentiality at layers 3, 4 and 7. Connectionless confidentiality also excludes layer 1, since there is no connectionless service at layer 1. Selective field confidentiality should only be supported by encipherment at layer 6 and invocation at layer 7, according to the semantics of the data.

Traffic flow confidentiality is recommended for layers 1, 3 and 7. Full traffic flow confidentiality is only possible at layer 1. This is achieved by the insertion of a pair of encipherment devices into the physical transmission path. Some of the effects of this mechanism may be produced by the use of a complete confidentiality service at one layer and the injection of spurious traffic at a higher layer. However, such a mechanism is costly and potentially consumes a large amount of resources. At layer 3, traffic padding may by used to provided limited confidentiality. At layer 7, the service is provided by the generation of spurious traffic in conjunction with confidentiality to prevent identification of the spurious traffic.

References

[1] IS 7498-2, "Information Processing Systems—Open Systems Interconnection—Security Architecture."

[2] IS 10021-1, "Information Processing Systems—Text Communication—MOTIS—Message Handling: System & Service Overview."

[3] IS 9594-8, "Information Processing Systems—Open Systems Interconnection—The Directory—Authentication Framework."

[4] IS 9797, "Data Cryptographic Techniques—Data Integrity Mechanism Using a Cryptographic Check."

[5] IS 8649, "ACSE Addenda on Authentication."

[6] SC21 N4207, "Authentication Framework. Draft Proposal."

[7] ISO 7498, "Information Processing Systems—Open Systems Interconnection—Basic Reference Model."

JAMES M. GALVIN is a Computer Scientist at Trusted Information Systems, Inc, in Glenwood, Maryland. Dr. Galvin's responsibilities emphasize communications security. He is Chair of the OSI Implementors' Workshop Security Special Interest Group, hosted by the National Institute of Standards and Technology. He is active in several working groups of the IETF, where he co-authored the SNMP Authentication and Privacy memos. He enjoys many fine lunches and dinners, but does not have a subscription to *Rolling Stone Magazine*.

Ed.: Dr. Galvin will chair a session on OSI Security at INTEROP 90.

User Viewpoint: The IP Security Option

by David Wiltzius, Lawrence Livermore National Laboratories

Introduction

As a Department of Energy (DoE) contractor, Lawrence Livermore National Laboratory (LLNL) must adhere to DoE computer security requirements. All components of the LLNL secure computer network are physically protected and isolated from uncleared personnel. Further, all data that passes over the network is physically protected at the highest data classification level on the network, to comply with what is known as the DoE's "system-high protection" requirement.

One of the chief priorities of LLNL's security strategy is the removal or minimization of chances for user error, such as mistyped addresses, to be misinterpreted as authorization to send data, for example, from a device approved for secret material to a device without that level of approval.

LLNL's network access parameters, in addition to the lab's use of a TCP/IP-like network protocol suite, create an optimal environment for a security structure modeled after the *Internet Protocol Security Option* (IPSO) [1].

RFC 1038

RFC 1038 defines a draft of the Basic IPSO for use by the Department of Defense (DoD). Additionally, it defines a draft of the *Extended IPSO* for use by any organization. To paraphrase the RFC's specifications, the Basic IPSO consists of a *type octet* (value 130) and a *length octet* plus two fields for the IPSO information (see Figure 1): the *security protection level* and the *protection authority flag*. The length octet specifies the number of octets for this option in the IP packet header and has a minimum value of 4. The security protection level field is an octet with an encoded value indicating the security at which the IP datagram should be protected. For example, the security protection level of "unclassified" is indicated by a value of 171 decimal (the security protection level values are defined such that errors occurring in packet delivery will not be likely to change its value to another valid security protection level).

Figure 1: Basic Security Option Format

The protection authority flag field is a variable number of octets. The first seven bits of each octet in this field indicate the *National Access Program(s)* or *Special Access Program(s)* whose rules apply to the protection of the IP datagram. The last bit in the octet is "0" if it is the final octet in the field, or "1" if there are additional octets. For example, a value of 144 (binary 10010000) is a protection authority flag field of length 1 indicating GENSER (as per DoD 5200.28) and the National Security Agency (NSA) as the protection authorities for the IP datagram.

Other security related information can be supplied in the *Extended IPSO*. This option (see Figure 2), also of variable length, consists of a *type octet* (value 133), a *length octet* (minimum value of 4) and two fields for the security information.

Extended IPSO

The Additional Security Info Format Code field is an octet whose value indicates the format and meaning of the *Additional Security Info* field. Security information is in the *Additional Security Info* field. If an organization wants to define an Extended IPSO it is required to:

- Get a value for the Additional Security Info Format Code assigned from the DCA,
- Define acceptance and rejection criteria of a datagram with this Extended IPSO. These criteria, unclassified if possible, will be made available from the DCA primarily so vendors can support that Extended IPSO.

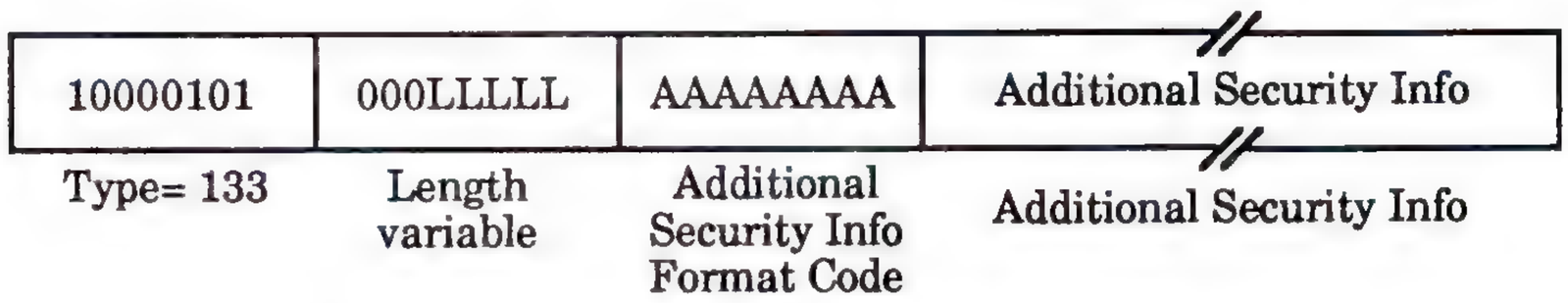

Figure 2: Extended Security Option Format

DoD Network Security Model

In addition to specifying the Basic and Extended IPSO, RFC 1038 also describes how the Basic IPSO is used to enforce the DoD network security requirements. Since RFC 1038 addresses many issues in detail regarding the enforcement of the DoD network security requirements, only a brief and partial overview will be provided here to give a flavor of the security model being used.

Each host on the network obtains a security level range and an authority protection flag at which it can operate. The host's security level range is assigned to it by one or more *protection authorities*. These protection authorities are then included in the host's authority protection flag.

Out-of-range procedure

For transmitting IP datagrams, the user specifies a security level and authority protection flag for the IP datagrams. The host's operating system must then validate these values prior to transmitting the IP datagram. That is, if the security protection level is not in the range assigned to the host, or if the authority protection flag is not a subset of the host's authority protection flag, then the values are invalid and an "out-of-range" procedure is followed. The "out-of-range" procedure overwrites the data in the IP datagram, issues the appropriate reply if a reply is permitted, and logs the out-of-range event, notifying a security officer as appropriate.

For received IP datagrams, the host's operating system performs a security check similar to the above on the values in the IP security option for both the security protection level and the authority protection flag.

Again, the above is only an overview of RFC 1038; most of the details in the RFC have been summarized or ignored. For example, DoE security personnel could authorize LLNL to use a particular computer for "confidential" through "secret" data. The operating system on that computer would then be given the security protection level range of "confidential" through "secret" and its authority protection flag would be set to the value 16 (binary 0000 1000) since DoE was the only protection authority in this example.

continued on next page

The IP Security Option *(continued)*

Implementation details

RFC 1038 does not specify any implementation details, but let's continue the example with a possible implementation using BSD *sockets*. An application transfers data by connecting a socket to the appropriate host. It then specifies the security protection level and authority protection flag for subsequent data by issuing a *setsockopt* with the appropriate arguments. This *setsockopt* call would return an error if the specified security protection level were not in the range assigned to this host, or if the authority protection flag in the *setsockopt* call had bits set that were not set in the host's authority protection flag.

If the *setsockopt* function call succeeded, then the user process can procede to transmit data and then every router that the IP datagram passed through would perform the same tests. That is, when the router receives an IP datagram it uses the security protection level range and authority protection flag associated with the network interface (and hence that network) to perform the security checks. Also, before transmitting the IP datagram onto the appropriate network, it uses the security protection level range and protection authority flag of the network interface (and hence that network) on which the IP datagram is to be transmitted to perform the security checks.

At this point, one may ask how the security protection level and protection authority flag for a network (and hence a network interface) are to be determined. One way may be to take the intersection of the security protection levels of all hosts on the network and the intersection (i.e., bitwise AND) of the authority protection flags of all hosts on the network. However, if the routers maintained security information for each host, it could be more selective in its enforcement of the IPSO.

The destination host then performs security checks on the IP datagrams it receives. It compares the security protection level in the IP datagram to its assigned security protection level range; if not in range the "out-of-range" procedure is followed. Additionally, each bit set in the authority protection flag of the IP datagram must also be set in the host's authority protection flag; if not, the "out-of-range" procedure is followed.

Delta-t

The security model at LLNL, which is almost identical to RFC 1038, is enforced by a combination of controls. LLNL has written and implemented its own network protocol, known as *Delta-t*. Delta-t uses a packet-framing format similar to that of TCP/IP, allowing LLNL to follow IPSO specifications. Mainframe operating systems designed by LLNL, in addition to kernel modifications in vendors' operating systems, *set* the required security level in outgoing packets as specified by the application. Routers and operating systems *check* the security level.

The classified network facility at LLNL is presently migrating to TCP/IP. Even though the Basic IPSO can be used in LLNL's security model, few implementations of TCP/IP sufficiently support this option. Furthermore, unless greater support for the IPSO is offered by vendors of TCP/IP products (and particularly host operating systems) before LLNL has completed its transition to TCP/IP, the Lab may be required to use a supplementary means of network security.

An effort is underway by the Department of Energy to use the Basic IPSO and to define an Extended IPSO. To allow DoE to use the Basic IPSO, bit 4 has been assigned for use in the Protection Authority Flags field. A proposal for a DoE Extended IPSO has also been drafted and is undergoing revision in the DoE community. It is the hope of the DoE representatives involved with this effort that, by providing a standard specification for the IPSO, vendors will be able to target a large enough audience to warrant the support of both the Basic and the DoE Extended IPSO.

Additionally, there is a set of security functions designed by Mitre for use on the DoDIIS (*DoD Intelligence Information System*) internet referred to as DNSIX (*DoDIIS Network Security for Information eXchange*). The three major areas of functionality defined by DNSIX are (1) session management for TCP sessions, (2) network audit, and (3) labeling for IP packets. DNSIX packet labeling is based upon the Basic IPSO to communicate the security level (Unclassified, Confidential, Secret, and Top Secret) of a packet across an IP-based network. DNSIX further defines variants of the Extended IPSO (whose content is undefined by RFC 1038) to communicate (1) a Session ID, or (2) a Network Level, which is a set of security compartments represented as a bit field.

Conclusion

The IPSO offers security potential to many organizations, particularly those involved with DoD or DoE contracts. Although it has been sadly under-implemented by vendors with only a few exceptions, it promises to have significant bearing on the future. For instance, several DoE contractors and research labs are moving to TCP/IP and have security needs similar to LLNL's. Furthermore, Version 2.0 of the *Government Open Systems Interconnection Profile* (GOSIP) specifies that the OSI security option will be composed of the fields in the Basic and Extended IP Security Options.

Organizations that plan to change from TCP/IP to OSI may retain a single security model from one stage to the next. Similarly, vendors that gain expertise in the IP Security Options can apply it to OSI networks. The current need of the IPSO standard by national laboratories and government contractors should offer a strong incentive to vendors to support the IPSO.

References

[1] St. Johns, M., "Draft revised IP security option," RFC 1038, January 1988.

[2] Brown, D. Private communications, May 17, 1990.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

[Ed.: This article is based on the author's article that appeared recently in the cisco Systems Customer Services Newsletter *The Packet*, Volume 2, No. 1., Spring 1990].

DAVE WILTZIUS leads the migration to TCP/IP effort at LLNL's Secure Computing Facility and heads the facility's Network Management team. He holds a BS and MS in Computer Science from the University of Wisconsin.

User Viewpoint: An Application using IP Packet Filtering

by Richard F. Kent, Network Systems Corporation

Introduction

Users of a large DoD IP internet have a unique requirement to have several large IBM hosts running TCP/IP appear as a single computing entity to the rest of the internet. This "single image" point of view would allow internet users to connect to a single IP address without knowledge of the actual IP address of the final host. The specific requirement is to allow FTP files to be sent to one IP address, and have some intelligent "front end" map each file stream to a system that is available for processing. This has to be done with off-the-shelf router hardware and software and without user software development. There are additional access control restrictions, but the main problem to be solved is the one-to-many relationship.

PCF

We attacked this requirement by recommending a mechanism called *Packet Control Facility* (PCF). PCF is a software filtering mechanism which allows IP routers to provide access control, packet filtering actions, type-of-service routing, and audit trail generation. PCF works by building a table in the IP router which filters datagrams based on header information and provides corresponding actions. For example, all packets sent to a particular address could be dropped, or all packets passing a particular filter test could have a copy sent to a third-party node, like a Network Management System. PCF is a general purpose software implementation which could run in any IP routing engine.

Proposed solution

Here is how it all works: All users direct incoming traffic to one single IP address (143.1.12.13 in the Figure 1 example). This is the address that is used for the single image perspective for all external hosts. Each connected host within the IBM complex is associated with two IP home addresses (i.e., it is *multihomed*). The first address is the single image address and the second is used to designate the individual host. All connected IBM hosts will recognize the common IP address, and individual IBM hosts will recognize their own IP address. PCF determines which "real hosts" receive packets that are addressed to the common IP address.

All incoming file streams are addressed to the common IP address. PCF determines on an individual stream basis which host receives the stream. The choice of which host to send the particular stream to can be based on a number of criteria like source address, type of service, or any filtering algorithm based on routing policy of choice. We decided on source IP address ranges for our implementation. In this case all incoming datagrams from a particular IP address range are forwarded to a predetermined destination host. Some obvious restrictions apply, for example, all packets from an individual stream must be sent to the same destination host. That is, a file may not be delivered in pieces to different hosts.

When the IBM hosts initiate transfers, the IBMs use their own individual IP address, so that PCF does not have to determine the difference between data transfers and ACKs. Any incoming datagram addressed to the individual host is an ACK and is allowed to pass through to the addressed host. By using this convention any IBM host can initiate a transfer to any host on the internet without worrying about another host receiving its acknowledgements.

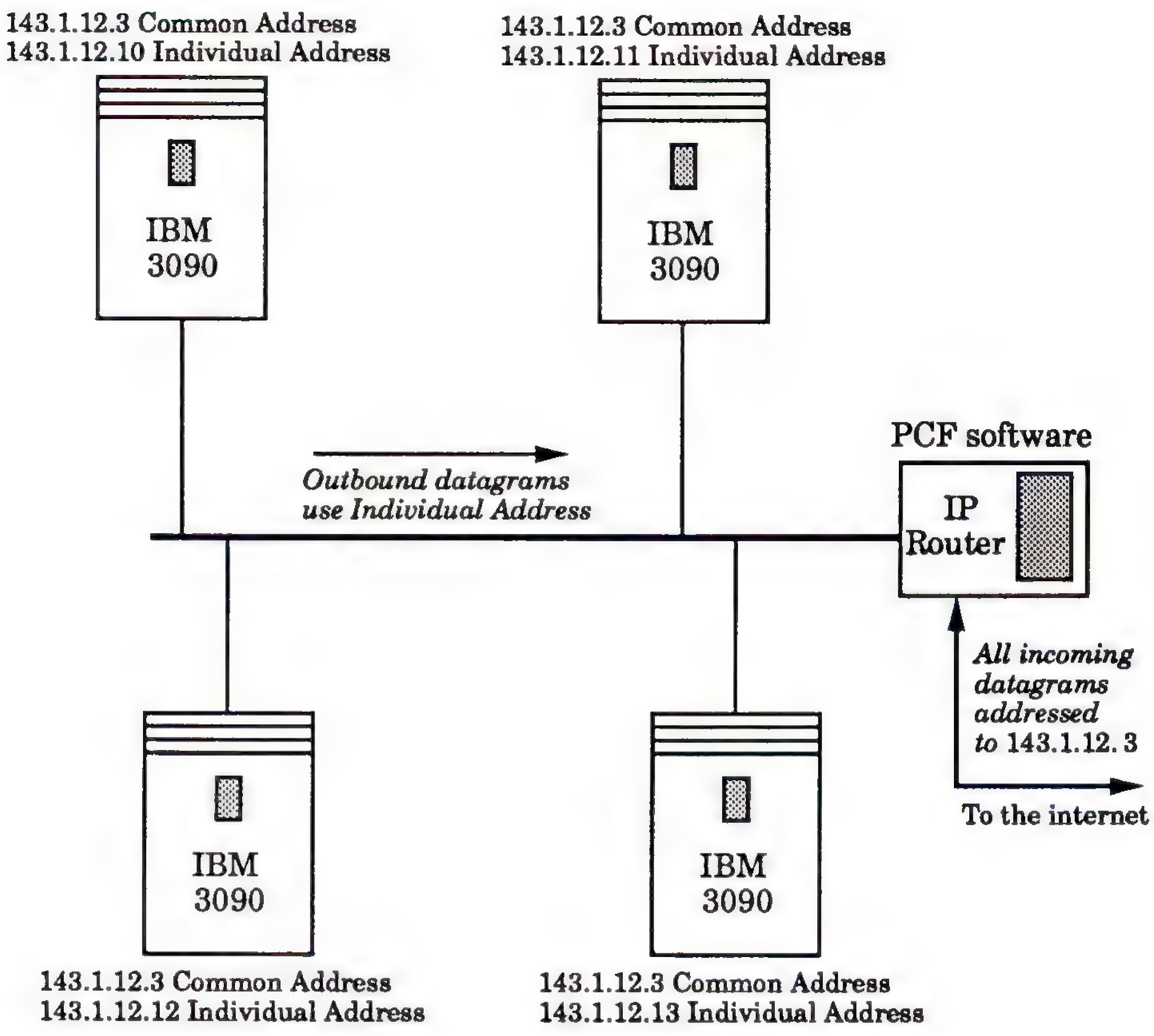

Figure 1: PCF Implementation

Additional benefits are derived in that if one of the IBM systems goes down PCF can choose to "rotor" to the next host for that stream. Currently new PCF filter table must be manually loaded when the system failure is detected; however this could be automated in a variety of ways.

Conclusion

PCF is a powerful tool that can be used to provide an effective single image perspective of hosts, provide an ability to route traffic around failed hosts, and enforce a security policy. Currently, load leveling algorithms must be statically determined; however, building static load leveling policies can provide a good balance of traffic among hosts. As technology advances in host software products and standard gateway protocols improve, both dynamic load leveling based on CPU utilization and fail-over techniques will become automated.

RICHARD F. KENT holds a B.S. in Computer Science from the University of Maryland. Rich, has been active in the computing industry since 1981. He has held a variety of technical positions in software development and management. He worked 4 years with Thomson CGR developing software for a variety of applications. He is currently engaged in the design, architecture, and systems engineering of large networks with Network Systems Corporation. Rich is also continuing his education towards an MBA at Loyola College in which he expects to complete in 1991.

The Computer Emergency Response Team

by Eileen Forrester, Carnegie Mellon University

Introduction

When a computer emergency occurs, often the greatest challenges for site managers are not technical, but communication and coordination problems among affected sites. To meet these challenges, the Internet community has formed *The Computer Emergency Response Team* (CERT). CERT is an informally organized group of experts that facilitates community response to computer security events involving Internet hosts.

After the Internet worm of November 1988, the *Defense Advanced Research Projects Agency* (DARPA) established the *CERT Coordination Center* (CERT/CC) at the SEI to improve communication during emergencies. The SEI was chosen as the home for the CERT/CC because it is uniquely positioned among the government, industry, and academic sites that are part of the Internet.

According to William Scherlis of DARPA, "The worm was a sad signal of the end of the era of widespread trust in the Internet community. The challenge we now face is to tighten security without compromising function, flexibility, interoperability, performance, and ease of access for researchers and other users—in other words, to maintain openness for exchange of scientific information and for growth in capability."

Remarking on the crisis that led to its creation, CERT/CC coordinator Rich Pethia said, "Events such as the Internet worm of November 1988 are unusual, but they serve as a warning that our increasing reliance on interconnected computers and networks creates new vulnerabilities."

Prevention and response

Pethia emphasizes proactive measures that can be taken by the CERT/CC and the Internet community to avoid security incidents. Scherlis confirms this: "The CERT has both prevention and response roles. Like a fire department, the response efforts are most widely visible; but, also like a fire department, the prevention efforts have the greatest long-term impact."

Because of media coverage of large-scale computer security incidents and the recent trial and conviction of the perpetrator of the Internet worm of November 1988, public attention has been focused on dramatic computer security problems. Less dramatic but more common events occur frequently and require just as effective responses. These events include intrusions of systems, as well as exploitations and discoveries of systems vulnerabilities.

Incidents

Since its inception in 1988, CERT/CC has responded to a continuous stream of reported security incidents. These include reports of intrusions, worms, and viruses, as well as reports of vulnerabilities and fixes for problems. At times, the CERT/CC has informed sites of intrusions before site administrators had themselves detected a problem. The majority of the incidents the CERT/CC responds to are due to lax password policies and failure to apply known fixes to security problems. Site managers can help to avoid security incidents by taking these key actions: Establish rigorous authentication policies for user access by providing password guidance to users and installing password filter programs to help users avoid passwords that can be easily cracked. Stay current with published security-related fixes.

Information clearinghouse

The CERT/CC offers assistance to members of the Internet community who wish to take further steps to heighten their awareness of security issues and increase the efficacy of their response to potential threats. The CERT/CC works with those who want to start their own CERT, according to Pethia. In addition, CERT/CC moderates several electronic mailing lists. These lists provide a forum for members of the community to exchange information about security issues, tools and systems, and viruses. CERT/CC also maintains online copies of publications about computer security produced by the National Institute of Standards and Technology, Computer Security Program Office.

Vendors

The CERT/CC works to increase security awareness among vendors as well as users. Increased communication provides advantages to both vendors and users. Vendors receive useful feedback from client communities and users are able to correct or work around dangerous security problems.

To handle computer security emergencies, CERT/CC provides a single point of contact for reporting incidents, 24 hours a day, 7 days a week. When an incident is reported, the CERT/CC works with CERT associates to determine the magnitude of the threat or problem. The CERT/CC then provides information to constituents on the nature of the problem and appropriate countermeasures to take.

Because it is the mission of the CERT system to enhance already existing security mechanisms, CERT organizations collaborate with other security organizations and pool resources when possible. The CERT system currently includes more than 600 contacts in industry, government, and the research community.

More information

To get further information or report problems, contact CERT/CC at the Internet address or the telephone numbers listed below.

CERT/CC Contact Information:

For emergencies: 412-268-7090

For information: 412-268-7080

FAX: 412-268-5758

Electronic mail: `CERT@sei.cmu.edu`

US mail:

CERT/CC
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

A previous version of this article appeared in the March 1990 issue of *Bridge*, a quarterly magazine published by the Software Engineering Institute (SEI). The SEI is a federally funded research and development center sponsored by the Department of Defense under contract to Carnegie Mellon University.

EILEEN FORRESTER is a technical writer and editor in the Information Management group at the SEI. She has a BA in English from Kent State University and is pursuing a Masters of Professional Writing from Carnegie Mellon.

CERT Advisories

Introduction

The CERT issues *Security Advisories* from time to time. The following article is a collection of hints for systems and network administrators taken from recent advisories.

Systems administrators should be aware that many systems around the Internet may could have security vulnerabilities which intruders know how to exploit. To avoid security breaches in the future, we recommend that all system administrators check for the kinds of problems noted in this article. This advisory describes problems with system configurations that we have seen intruders using. In particular, intruders have attempted to exploit problems in Berkeley BSD derived UNIX systems and have attacked DEC VMS systems. In the advisory below, points 1 through 12 deal with UNIX, points 13 and 14 deal with the VMS attacks. If you have questions about a particular problem, please get in touch with your vendor.

We've had reports of intruders attempting to exploit the following areas:

Passwords

1) Use of TFTP (*Trivial File Transfer Protocol*) to steal password files: To test your system for this vulnerability, connect to your system using TFTP and try "`get /etc/motd`." If you can do this, anyone else can get your password file as well. To avoid this problem, disable *tftpd*.

In conjunction with this, encourage your users to choose passwords that are difficult to guess. Furthermore, inform your users not to leave any clear text username/password information in files on any system. If an intruder can get a password file, he/she will usually take it to another machine and run password guessing programs on it. These programs involve large dictionary searches and run quickly even on slow machines. The experience of many sites is that most systems that do not put any controls on the types of passwords used probably have at least one password that can be guessed.

2) Exploiting accounts without passwords or known passwords (accounts with vendor supplied default passwords are favorites). Also uses *finger* to get account names and then tries simple passwords: Scan your password file for extra UID 0 accounts, accounts with no password, or new entries in the password file. Always change vendor supplied default passwords when you install new system software.

Sendmail

3) Exploiting holes in *sendmail:* Make sure you are running the latest *sendmail* from your vendor. BSD 5.61 fixes all known holes that intruders have been using.

FTP

4) Exploiting bugs in old versions of FTP; exploit mis-configured anonymous FTP: Make sure you are running the most recent version of FTP which is the Berkeley version 4.163 of Nov. 8 1988. Check with your vendor for information on configuration upgrades. Also check your anonymous FTP configuration. It is important to follow the instructions provided with the operating system to properly configure the files available through anonymous ftp (e.g., file permissions, ownership, group, etc.). Note especially that you should not use your system's standard password file as the password file for FTP.

Finger

5) Exploiting the *fingerd* hole used by the Morris *Internet Worm*. Make sure you're running a recent version of *finger*. Numerous Berkeley BSD derived versions of UNIX could be vulnerable.

Some other things to check for:

rhosts

6) Check user's .rhosts files and the `/etc/hosts.equiv` files for systems outside your domain. Make sure all hosts in these files are authorized and that the files are not world-writable.

cron* and *at

7) Examine all the files that are run by *cron* and *at*. We've seen intruders leave back doors in files run from *cron* or submitted to *at*. These techniques can let the intruder back on the system even after you've kicked him/her off. Also, verify that all files/programs referenced (directly or indirectly) by the *cron* and *at* jobs, and the job files themselves, are not world-writable.

uucp

8) If your machine supports *uucp*, check the `L.cmds` file to see if they've added extra commands and that it is owned by root (not by uucp!) and world-readable. Also, the `L.sys` file should not be world-readable or world-writable.

Mail aliases

9) Examine the `/usr/lib/aliases`(mail alias) file for unauthorized entries. Some alias files include an alias named "`uudecode`;" if this alias exists on your system, and you are not explicitly using it, then it should be removed.

Hidden files

10) Look for hidden files (files that start with a period and are normally not shown by *ls*) with odd names and/or setuid capabilities, as these can be used to "hide" information or privileged (setuid root) programs, including `/bin/sh`. Names such as "`..` " (dot dot space space), "`...`", and `.xx` have been used, as have ordinary looking names such as "`.mail`." Places to look include especially `/tmp`, `/usr/tmp`, and hidden directories (frequently within users' home directories).

System programs

11) Check the integrity of critical system programs such as *su, login,* and *telnet*. Use a known, good copy of the program, such as the original distribution media and compare it with the program you are running.

12) Older versions of systems often have security vulnerabilities that are well known to intruders. One of the best defenses against problems is to upgrade to the latest version of your vendor's system.

VMS vulnerabilities

13) Intruder may use system default passwords that have not been changed since installation. Make sure to change *all* default passwords when the software is installed. Intruders also try to guess simple user passwords.

14) If an intruder gets into a system, often the programs `loginout.exe` and `show.exe` could be modified. Check these programs against the files found in your distribution media.

Ed.: There will be several sessions on Security at INTEROP® 90. Call us at 1-800-INTEROP or 415-941-3399 to receive the INTEROP 90 Advance Program.

UNIX Security white paper available

Focus

A new white paper from SRI International's Information and Telecommunication Sciences and Technology Division is now available. The paper, "Improving the Security of Your UNIX System," describes measures that you as a system administrator can take to make your UNIX system(s) more secure. Oriented primarily at SunOS 4.x, most of the information covered applies equally well to any Berkeley UNIX system with or without NFS and/or Yellow Pages (NIS). Some of the information can also be applied to System V, although this is not a primary focus of the paper.

Format

In order to format the paper, the *troff* text formatter and the "-ms" macro package (available with any Sun or Berkeley UNIX system) are required. You *do not* need a *PostScript* printer, unless you want to print the cover page with the SRI logo on it.

Getting the paper

The paper is available from the host `spam.istd.sri.com` (128.18.4.3) via anonymous FTP. File: `pub/security-doc.tar.Z`. Be sure to remember to set "image" mode on the transfer. Sorry, UUCP access is not available—if you don't have Internet access, find a friend who does.

—Dave Curry

Mailing list for Privacy Enhanced Mail

A new mailing list, PEM-DEV, has been created for discussions related to the development and deployment of *Privacy Enhanced Mail* systems based on Internet RFCs 1113, 1114, and 1115.

RFC 1113–15

RFC 1113 specifies protocol extensions and processing procedures for cryptographic-based message encipherment and authentication for Internet electronic mail. RFC 1114 specifies a supporting key management architecture and infrastructure, based on public key certificates. The key management architecure is compatible with the authentication framework described in CCITT X.509. RFC 1115 describes algorithm and related information relevant to the other RFCs.

TIS, BBN, RSADSI and NIST are jointly developing RFC compliant software for deployment in the Internet in order to encourage the use and development of RFC-based privacy enhanced mail systems. Beta testing of the software will occur this spring and release to the full Internet is anticipated this summer.

Topics

The PEM-DEV mailing list is intended to cover a wide range of issues including:

- Issues related to the protocol extensions and message processing procedures and the key management architecture and infrastructure specified in the RFCs, including questions and answers, clarification of details, unpublished changes, etc.
- Issues related to the development and deployment of Privacy Enhanced Mail systems, including technical issues, development status, availability, etc.

Please send contributions to the list proper to: `pem-dev@tis.com`. Administrivia, e.g., additions to or deletions from the list, should be sent to: `pem-dev-request@tis.com`.

—David Balenson

FIPS documents on security available online

Five *Federal Information Processing Standards* (FIPS) publications are available online at the *Security Coordination Center* (SCC). These documents address various aspects of computer and network security. FIPS 500-166 is a guide regarding managing the threat of computer viruses or intrusions. Also available are the executive, management, and user guides regarding methods to protect information resources.

Getting the files

You can obtain these files on the `nic.ddn.mil` host [192.67.67.20] via anonymous FTP or Kermit. You can also get them via e-mail by sending a message to `Service@nic.ddn.mil` with the subject line reading "Send SCC:FIPS_500_###.TXT" (replace the ### with the number of the file you want and end with .PS instead of .TXT for the *PostScript* version of FIPS 500-166). If you have questions about the procedures for obtaining these files, please call 1-800-235-3155 or send a message to `SCC@nic.ddn.mil`

Below are the file names as they are online and the descriptive titles of the documents. Use the file name, rather than the document title, when requesting the file online. All files are in the "SCC:" directory.

File Name:	Document Title:
FIPS_500_166.TXT (or PS)	*Computer Viruses and Related Threats: A Management Guide*
FIPS_500_169.TXT	*Executive Guide to the Protection of Information Resources*
FIPS_500_170.TXT	*Management Guide to the Protection of Information Resources*
FIPS_500_171.TXT	*Computer User's Guide to the Protection of Information Resources*

For further reading

Network Management and Security has been covered extensively in *ConneXions* in the past. Some highlights include:

ConneXions Volume 3, No. 3, March 1989: *Special issue on Network Management*.

Neumann, P., "Long-Term Implications of the Internet Worm," *ConneXions,* Volume 3, No. 4, April 1989.

Ostapik, F., "Effect of the Internet Worm on Security," *ConneXions,* Volume 3, No. 9, September 1989.

Ostapik, F., and Marine, A., "The CERT Workshop," *ConneXions,* Volume 3, No. 11, November 1989.

Schiller, J., "Kerberos: Network Authentication for Today's Open Networks," *ConneXions,* Volume 4, No. 1, January 1990.

Postel, J. "Book Review: *The Cuckoo's Egg," ConneXions,* Volume 4, No. 1, January 1990.

Dern, D., "Interview with Steve Kent on Internet Security," *ConneXions,* Volume 4, No. 2, February 1990.

Dern, D. "The Trusted Mail System," *ConneXions,* Volume 4, No. 2, February 1990.

Upcoming Events

LISA

The *Fourth USENIX Large Installation Systems Administration* (LISA) conference will be held in Colorado Springs, Colorado on October 18–19, 1990. A tutorial program will be offered in conjunction with the workshop on October 17.

The program committee will be reviewing papers submitted on subjects including but not limited to:

- Automation of tasks
- Network management
- Distributed services
- System backup
- File and data archiving
- Electronic mail
- Security
- Account/user management
- Accounting
- USENET News/Notes
- Performance monitoring & tuning
- Configuration management
- Vendor issues
- Distributed administration

For further information, contact the program chair:

Steve Simmons
Industrial Technology Institute
2901 Hubbard Road
Ann Arbor, MI. 48109
313-769-4086
scs@iti.org

UNIX Security Workshop

The *Second USENIX UNIX Security Workshop* will be held at the Marriott Hotel in Portland, Oregon, August 27–28, 1990. It will bring together researchers in computer security dealing with UNIX and system administrators trying to use UNIX in environments where protection and security are of vital importance. It is intended to provide an environment where researchers can discuss their latest results, where researchers and practitioners can discuss the applicability of those results to practical problems, and where system administrators can share their unique solutions for dealing with problems.

The issues covered by this workshop include both theoretical topics and everyday problems. Some topics to be considered include:

- Modeling the UNIX operating system theoretically
- Password security
- Network security
- Security in a distributed system or environment
- File system security
- Computer worms, viruses, and other phenomena
- New designs to obtain C-level (or better) certification
- Making existing UNIX systems more secure

For further information, contact the conference chair:

Matt Bishop
Dept. of Mathematics and Computer Science
Bradley Hall
Dartmouth College
Hanover, NH 03755
603-646-3267
decvax!dartvax!Matt.Bishop
Matt.Bishop@dartmouth.edu

For more information about these or other USENIX conferences, contact the USENIX office: office@usenix.org, or 415-528-8649.

Call for Participation

A *Workshop on Distributed Systems: Operations & Management* will be held October 22–23, 1990, in Berlin, Germany. The workshop is sponsored by IFIP TC 6 WG 6 on Network Management with participation of the IEEE Communications Society/CNOM.

Topics

All aspects of operations and management of distributed computing, including integration and interplay of and among systems management functions (fault, configuration, accounting, performance, and security management) and distributed operating systems, will be discussed.

Distributed/Network computing, currently 50% of all computing, is projected to dramatically increase to 76% by 1992 (Sanford C. Bernstein). Distributed systems are characterized by complexity, large numbers of diverse components and multiple autonomous processes requiring sophisticated management. This dramatic growth coupled with the complexity of such systems necessitates new solutions. This workshop will address areas related to operations and management of distributed computing. It will bring together an international, expert group of researchers, system integrators, participants in standardization efforts, vendor representatives, and users.

The workshop is intended to foster cooperation among people actively working in this area. To encourage discussion, attendance will be limited to 40 participants. Selection of participants will be by invitation primarily based on contributions. Selection of contributions will be based on single page abstracts.

Venue

The workshop will be held at the GMD-FIRST location in Berlin, Germany October 22–23, 1990. It is planned for one and a half days. Immediately following the Workshop, the regular meeting of the IFIP WG 6.6 and the Program Committee meeting of the Second International Symposium on Integrated Network Management will occur.

If you would like to participate and/or contribute please mail your response to Branislav Meandzija at `meandzij@ucrmath.ucr.edu`.

Important dates

Abstract due:	Immediately
Intention to participate:	Immediately
Tentative program mailed:	September 15, 1990

For more information contact:

Wolfgang Zimmer
GMD-FIRST
Hardenbergplatz 2
D-1000 Berlin 12
West Germany

or

Branislav Meandzija
Department of Computer Science
University of California
Riverside, CA 92521-0135
U.S.A.

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

Bulk Rate
U.S. POSTAGE
P A I D
SAN JOSE, CA
PERMIT NO. 1